

# **Tekijänoikeus osana kokeiluja hyödyntävän ohjelmistoyrityksen IPR-strategiaa**

Ilona Saarenpää  
Lapin yliopisto  
Pro gradu -tutkielma  
Oikeustieteiden tiedekunta  
Immateriaalioikeus  
2018

# SISÄLLYS

<b>Lähteet</b> .....	III
Kirjallisuus .....	III
Virallislähteet .....	X
Oikeuskäytäntö.....	XI
Tekijänoikeusneuvoston lausunnot .....	XII
Muut lähteet .....	XIII
<b>Lyhenteet</b> .....	XV
<b>1. JOHDANTO</b> .....	1
1.1 Johdatus aiheeseen .....	1
1.2 Tutkimuskysymykset ja rajaus.....	3
1.3 Metodi .....	4
1.4 Tutkielman rakenne.....	5
<b>2. KOKEILUT INNOVAATIOPROSESSISSA</b> .....	6
2.1 Kokeilut, kokeilukulttuuri ja kokeiluyhteiskunta.....	6
2.2 Suunnittelemisen ja kokeileminen.....	7
2.3 Kokeilu syklinä .....	8
2.3.1 Kokeilun vaiheet .....	8
2.3.2 Kokeilun suunnittelu.....	9
2.3.3 Kokeiluasetelman valmistelu .....	9
2.3.4 Kokeilun suorittaminen, tulosten analysointi ja jatkotoimista päättäminen ...	11
<b>3. IMMATERIAALIOIKEUDET OSANA YRITYKSEN STRATEGIAA</b> .....	12
3.1 IPR-strategian käsite .....	12
3.2 IPR-strategian elementtejä.....	13
3.3 Tekijänoikeuden erityispiirteitä IPR-strategian kannalta .....	15
<b>4. KOKEILUASETELMA TEKIJÄNOIKEUDELLISESTA NÄKÖKULMASTA</b> .....	17
4.1 Tekijänoikeuden oikeuttamisperusteista .....	17
4.2 Tekijänoikeuden kohde .....	19
4.2.1 Teostaso .....	19
4.2.2 Muodon ja idean dikotomia .....	20
4.3 Tietokoneohjelman prototyyppi teoksena .....	22
4.3.1 Ohjelmakoodi ja tekijänoikeus.....	22
4.3.2 Rajapinnat ja tekijänoikeus.....	25
4.4 Kokeilutilanne ja teoksen laillinen käyttäminen.....	27
4.4.1 Taloudelliset ja moraaliset oikeudet .....	27
4.4.2 Jälkiperaisteokset .....	30
<b>5. NOPEA JA EDULLINEN PROTOTYPOINTI TEKIJÄNOIKEUDELLISENA HAASTEENA</b> .....	32
5.1 Valmiin materiaalin lisensointi .....	32
5.2 Prototyyppi ilman koodaamista.....	34
5.3 Komponenttipohjainen ohjelmointi.....	34
5.4 Avoimen lähdekoodin hyödyntäminen .....	35
5.4.1 Mitä avoin lähdekoodi on? .....	35
5.4.2 Sallivat lisenssit .....	37

5.4.3 Vastavuoroisuutta vaativat lisenssit .....	39
5.4.4 Lisenssien yhteensopivuus.....	42
5.4.5 Lisenssien vastuunrajoituslausekkeet .....	45
5.5 Strategisia näkökohtia .....	47
<b>6. TEKIJYYS JA TEKIJÄNOIKEUDEN SIIRTYMINEN KOKEILUSSA.....</b>	<b>49</b>
6.1 Tekijänoikeuden subjekti ja tekijänoikeudellinen yhteistyö.....	49
6.2 Tekijyyden määrittäminen kokeilussa.....	51
6.2.1 Koehenkilöiden ja kehitystiimin rooleista .....	51
6.2.2 Tekijänoikeudellisesti relevantti toiminta.....	53
6.2.3 Voiko koehenkilö olla tekijä? .....	54
6.3 Oikeuksien siirtyminen.....	56
6.3.1 Kokeiluun osallistuvien henkilöiden ja yrityksen välinen oikeussuhde.....	56
6.3.2 Oikeuksien siirtäminen sopimalla .....	58
6.3.3 Oikeuksien siirtyminen työsuhteen perusteella .....	59
6.4 Strategisia näkökohtia .....	61
<b>7. TEKIJÄNOIKEUS, MUUT SUOJAKEINOT JA NIIDEN YHTEISKÄYTTÖ .....</b>	<b>64</b>
7.1 Patentti.....	64
7.1.1 Tietokoneohjelmalla toteutetun ratkaisun patentoitavuus .....	64
7.1.2 Keksinnön uutuuden säilyttäminen kokeilussa .....	65
7.1.3 Tekijänoikeus ja patentti kokeilussa: vertailua.....	66
7.2 Ei-muodolliset suojauskeinot kokeilussa tekijänoikeuden rinnalla.....	68
7.3 Avoin innovaatio .....	71
7.3.1. Mitä avoimuudella tarkoitetaan?.....	71
7.3.2 Avoin innovaatio ja suojakeinot.....	72
7.3.3 Avoimuus kokeilussa.....	73
<b>8. LOPPUPÄÄTELMÄT .....</b>	<b>75</b>

# LÄHTEET

## Kirjallisuus

- ABID, JACK GEORGE (2005) Software Patents on Both Sides of the Atlantic. *John Marshall Journal of Computer & Information Law*, 23, Summer 2005, s. 815–844.
- ALEXY, OLIVER – CRISCUOLO, PAOLA – SALTER, AMMON (2009) Does IP Strategy Have to Cripple Open Innovation? *MIT Sloan Management Review*, 51(1), s. 71–77.
- ALMEIDA, DANIEL A. – MURPHY, GAIL C. – WILSON, GREG – HOYE, MIKE (2017) Do Software Developers Understand Open Source Licenses? *Proceedings of the 25th International Conference on Program Comprehension*, s. 1–11.
- ANDERSON, ERIC T. – SIMESTER, DUNCAN (2011) A Step-by-step Guide to Smart Business Experiments. *Harvard Business Review*, March 1/2011, s. 98–105.
- ASHORN, TIINA (2003) Immateriaalioikeudet yrityksen liiketoimintastrategiana. Teoksessa Niinikoski, Marja-Liisa – Sibelius, Kaisa (toim.): *Kulttuuribusiness*, s. 84–91. Helsinki: WSOY.
- BADAMPUDI, DEEPIKA – WOHLIN, CLAES – PETERSEN, KAI (2016) Software Component Decision-making: In-house, OSS, COTS or Outsourcing – A Systematic Literature Review. *The Journal of Systems and Software*, 121, 2016, s. 105–124.
- BAYERSDORFER, MITCH (2007) Managing a Project with Open Source Components. *Interactions*, 14(6), 2007, s. 33–34.
- BERG, ANNUKKA – HILDÉN, MIKAEL – LAHTI, KIRSI (2014) *Kohti kokeilukulttuuria. Analyysi Jyväskylän resurssiviisaista kokeiluista strategisen kehittämisen työkaluina*. Sitran selvityksiä 77. Helsinki.
- BERG, ANNUKKA (2013) *Kokeilun paikka! Suomi matkalla kohti kokeiluyhteiskuntaa*. Helsinki: Eduskunnan tulevaisuusvaliokunta.
- BREM, ALEXANDER – NYLUND, PETRA – HITCHEN, EMMA (2017) Open Innovation and Intellectual Property Rights: How Do SMEs Benefit from Patents, Industrial Designs, Trademarks and Copyrights? *Management Decision*, 55(6), s. 1285–1306.
- CARVER, BRIAN W. (2005) Share and Share Alike: Understanding and Enforcing Open

- Source and Free Software Licenses. *Berkeley Technology Law Journal*, 20(1), s. 443–481.
- CHESBROUGH, HENRY (2006) *Open Innovation. The New Imperative for Creating and Profiting from Technology*. Boston: Harvard Business School Press.
- CHESBROUGH, HENRY – APPLEYARD, MELISSA (2007) Open Innovation and Strategy. *California Management Review*, 50(1), 2007, s. 57–76.
- DAVENPORT, THOMAS H. (2009) How to Design Smart Business Experiments. *Harvard Business Review*, February 2009, s. 68–76.
- DETERMANN, LOTHAR (2006) Dangerous Liaisons – Software Combinations as Derivative Works? Distribution, Installation, and Execution of Linked Programs Under Copyright Law, Commercial Licenses, and the GPL. *Berkeley Technology Law Journal*, 21(4), Fall 2006, s. 1421–1498.
- DI GIACOMO, PIERGIORGIO (2005) COTS and Open Source Software Components: Are They Really Different on the Battlefield? *Proceedings of the International Conference on COTS-Based Software Systems, ICCBSS 2005*, s. 301–310.
- DIXON, ROD (2004) *Open Source Software Law*. Boston: Artech House.
- EDELMAN, TOM (2012) Lisenssisopimus. Teoksessa Saarnilehto, Ari – Annola, Vesa – Hemmo, Mika – Karhu, Juha – Kartio, Leena – Tammi-Salminen, Eva – Tolonen, Juha – Tuomisto, Jarmo – Viljanen, Mika: *Varallisuusoiikeus*. 2. uudistettu painos. Helsinki: Sanoma Pro, s. 1097–1105.
- EVANS, DAVID – LAYNE-FARRAR, ANNE (2004) Software Patents and Open Source: The Battle Over Intellectual Property Rights. *Virginia Journal of Law & Technology*, 9(10), s. 1–27.
- FELIN, TEPPU – ZENGER, TODD (2014) Closed or Open Innovation? Problem Solving and the Governance Choice. *Research Policy*, 43, 2014, s. 914–925.
- GERMAN, DANIEL M. – DI PENTA, MASSIMILIANO – GUEHENEUC, YANN-GAEL – ANTONIOL, GIULIANO (2009) Code Siblings: Technical and Legal Implications of Copying Code between Applications. *Proceedings of the 6th International Working Conference on Mining Software Repositories*, 2009, s. 81–90.
- GERMAN, DANIEL M. – HASSAN, AHMED E. (2009) License Integration Patterns: Address-

- ing License Mismatches in Component-Based Development. *Proceedings of the 31st International Conference on Software Engineering, ICSE 2009*, s. 188–198.
- GHAREHYAZIE, MOHAMMAD – RAY, BAISHAKHI – FILKOV, VLADIMIR (2017) Some from Here, Some from There: Cross-Project Code Reuse in GitHub. *Proceedings of the 14th International Conference on Mining Software Repositories*, s. 291–301.
- GOLDSTEIN, PAUL – HUGENHOLTZ, BERNT (2013) *International Copyright. Principles, Law, and Practice*. 3rd edition. New York: Oxford University Press.
- GOLLIN, MICHAEL (2008) *Driving Innovation – Intellectual Property Strategies for a Dynamic World*. Cambridge: Cambridge University Press.
- GRÖNLUND, MIKKO – PÖNNI, VEIJO – TOIVONEN, TIMO E. – SINERVO, PETTERI (2014) *Economic Contribution of Copyright-based Industries in Finland 2009–2012*. Helsinki: Suomen tekijänoikeudellinen yhdistys ry.
- HAARMANN, PIRKKO-LIISA (2014) *Immateriaalioikeus*. Helsinki: Talentum.
- HAARMANN, PIRKKO-LIISA (1992) *Tekijänoikeus, lähioikeudet ja oikeus valokuvaan*. Helsinki: Lakimiesliiton kustannus.
- HAARMANN, PIRKKO-LIISA – MANSALA, MARJA-LEENA (2012) *Immateriaalioikeuden perusteet*. Helsinki: Talentum.
- HALL, BRONWYN – HELMERS, CHRISTIAN – ROGERS, MARK – SENA, VANIA (2014) The Choice between Formal and Informal Intellectual Property: A Review. *Journal of Economic Literature*, 52(2), s. 375–423.
- HAMMOUDA, IMED – MIKKONEN, TOMMI – OKSANEN, VILLE – JAAKSI, ARI (2010) Open Source Legality Patterns: Architectural Design Decisions Motivated by Legal Concerns. *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek '10*, s. 207–214.
- HARENKO, KRISTIINA – NIIRANEN, VALTTERI – TARKELA, PEKKA (2016) *Tekijänoikeus*. Helsinki: Talentum Pro.
- HASSI, LOTTA – PAJU, SAMI – MAILA, REETTA (2015) *Kehitä kokeillen. Organisaation käsikirja*. Helsinki: Talentum Pro.

- HASSI, LOTTA – TUULENMÄKI, ANSSI (2012) Experimentation-driven Approach to Innovation: Developing Novel Offerings Through Experiments. *Proceedings of the XXIII ISPIM-conference*, s. 1–13.
- HENKEL, JOACHIM – SCHÖBERL, SIMONE – ALEX, OLIVER (2014) The Emergence of Openness: How and Why Firms Adopt Selective Revealing in Open Innovation. *Research Policy*, 43, 2014, s. 879–890.
- HIETANEN, HERKKO (2008) *The Pursuit of Efficient Copyright Licensing. How Some Rights Reserved Attempts to Solve the Problems of All Rights Reserved*. Acta Universitatis Lappeenrantaensis 523.
- HOLGERSSON, MARCUS – GRANSTRAND, OVE (2017) Patenting Motives, Technology Strategies, and Open Innovation. *Management Decision*, 55(6), s. 1265–1284.
- HOLGERSSON, MARCUS – WALLIN, MARTIN (2017) The Patent Management Trichotomy: Patenting, Publishing, and Secrecy. *Management Decision*, 55(6), s. 1087–1099.
- HONKINEN, TUOMAS – INNANEN, ANTTI – LINDGREN, JAAKKO – PELLO, JUKKA – RANTANEN, JOHANNA – SILTALA, KYÖSTI – TUOMALA, SARA (2016) *Startup-juridiikan käsikirja*. Helsinki: Alma Talent.
- INNANEN, ANTTI – JÄSKE, JUKKA (2014) *Brändin suoja*. Helsinki: Edita.
- JANSEN, SLINGER – BRINKKEMPER, SJAAP – HUNINK, IVO – DEMIR, CETIN (2008) Pragmatic and Opportunistic Reuse in Innovative Start-up Companies. *IEEE Software*, 25(6), Nov/Dec 2008, s. 42–49.
- KIVIMÄKI, T.M. (1966) *Uudet tekijänoikeus- ja valokuvauslait*. Helsinki: WSOY.
- KOILLINEN, MIKAEL – LAVAPURO, JUHA (2002) Tekijänoikeudet tietoyhteiskunnassa perusoikeusnäkökulmasta. Teoksessa Kulla, Heikki (toim.): *Viestintäoikeus*. Helsinki: WSOY, s. 335–355. (e-kirja)
- KOIVUMAA, ARI (1995) *Tekijänoikeus informaatioyhteiskunnassa – Tietokoneohjelman tekijöiden oikeudellinen suoja*. Rovaniemi.
- LAHTI, ARTO (2008) *Ohjelmistoteollisuuden globaali kasvustrategia ja immateriaalioikeudet*. Helsinki: Helsinki School of Economics.

- LAHTI, KIM (2006) *Teknologiayritysten globaali kasvustrategia ja immateriaalioikeudet*. Helsinki: Tekniikan akateemisten liitto TEK, International Networking Publishing.
- LAVAPURO, JUHA (2007) Lähtökohtia perusoikeusmyönteiselle tekijänoikeudelle. Teoksessa Mylly, Tuomas – Lavapuro, Juha – Karo, Marko (toim.): *Tekemisen vapaus – Luovuuden ehdot ja tekijänoikeus*. Helsinki: Gaudeamus, s. 148–170.
- LEPPÄMÄKI, LAURA (2007) Tekijänoikeudeton alue: public domain. Teoksessa Mylly, Tuomas – Lavapuro, Juha – Karo, Marko (toim.): *Tekemisen vapaus – Luovuuden ehdot ja tekijänoikeus*. Helsinki: Gaudeamus, s. 80–93.
- LESSIG, LAWRENCE (2001) *The Future of Ideas. The Fate of the Commons in a Connected World*. New York: Vintage.
- LI, JINGYUE – CONRADI, REIDAR – SLYNGSTAD, ODD PETTER N. – BUNSE, CHRISTIAN – TORCHIANO, MARCO – MORISIO, MAURIZIO (2006) An Empirical Study on Decision Making in Off-The-Shelf Component-Based Development. *Proceedings of the 28th International Conference on Software Engineering*, ACM, s. 897–900.
- LINÅKER, J. – MUNIR, H. – WNUK, K. – MOLS, C.E. (2017) Motivating the Contributions: An Open Innovation Perspective on What to Share as Open Source Software. *The Journal of Systems and Software*, 135, 2018, s. 17–36.
- LOCKE, JOHN (1982) *Second Treatise of Government*. Arlington Heights: Harlan Davidson.
- LOHI, MARKUS (2009) Tekijänoikeusluovutuksen laajuus ja tulkintaperiaatteet työsuhteessa – erityisesti uusia käyttötapoja silmällä pitäen. Teoksessa Oesch, Rainer – Heiskanen, Hanna – Hyyrynen, Outi (toim.): *Tekijänoikeus ja digitaalitalous*. Helsinki: WSOY, s. 31–59.
- LUOMA, RAIMO – HUUSKONEN, MIKKO (2008) *IPR tehokkaaseen käyttöön! Aineksia teollis- ja tekijänoikeuksien strategiaan*. Työ- ja elinkeinoministeriön julkaisuja 37/2008. Helsinki: Työ- ja elinkeinoministeriö.
- MANSALA, MARJA-LEENA (2001) Immateriaalioikeuden lisensiointi. Teoksessa Mylly, Tuomas (toim.): *Immateriaalioikeudet kansainvälisessä kaupassa*. Helsinki: Kauppa-kaari, s. 1–29.



- MANSALA, MARJA-LEENA (2007) Pienyrityksen immateriaalioikeuksien suojaamisstrategia. *Defensor Legis* 4/2005, s. 587–601.
- MOULIN, AXEL – THUE LIE, HAAKON (2005) *Intellectual Property Rights and Nordic SMEs*. Oslo: Leogriff AS.
- MYLLY, TUOMAS (2004) Tekijänoikeuden ideologiat ja myytit. *Lakimies* 2/2004, s. 228–254.
- MYLLY, ULLA-MAIJA (2005) Tietokoneohjelmien rajapintojen tekijänoikeussuoja. *Lakimies* 5/2005, s. 746–767.
- NCUBE, CORNELIUS – OBERNDORF, PATRICIA – KARK, ANATOL W. (2008) Opportunistic Software Systems Development: Making Systems from What's Available. *IEEE Software*, 25(6), Nov/Dec 2008, s. 38–41.
- NGUYEN-DUC, ANH – WANG, XIAOFENG – ABRAHAMSSON, PEKKA (2017) What Influences the Speed of Prototyping? An Empirical Investigation of Twenty Software Start-ups. *Proceedings of the 18th International Conference on Agile Software Development, XP 2017*, s. 20–36.
- OESCH, RAINER (2005) Tekijänoikeudet ja perusoikeusnäkökulma. *Lakimies* 3/2005, s. 351–376.
- OESCH, RAINER (2017) Johdatus aiheeseen: yleinen etu ja immateriaalioikeuden suoja – mitä uutta? Teoksessa Oesch, Rainer – Eloranta, Mikko – Heino, Mari – Kokko, Mira (toim.): *Immateriaalioikeudet ja yleinen etu*. Helsinki: Alma Talent, s. 1–19.
- OESCH, RAINER – PIHLAJAMAA, HELI – SUNILA, SAMI (2014) *Patenttioikeus*. Helsinki: Talentum.
- PALFREY, JOHN (2012) *Intellectual Property Strategy*. Cambridge: MIT Press.
- PIHLAJARINNE, TAINA (2014) Immateriaalioikeudet. Teoksessa Villa, Seppo – Airaksinen, Manne – Bärlund, Johan – Jauhiainen, Jyrki – Kaisanlahti, Timo – Knuts, Mårten – Kuoppamäki, Petri – Mähönen, Jukka – Pihlajarinne, Taina – Raitio, Juha – Rissanen, Kirsti – Viitanen, Klaus – Wilhelmsson, Thomas: *Yritysoikeus*. Helsinki: Talentum, s. 857–972.
- PITKÄNEN, ANU (2016) *Tavaramerkkioikeuden ja tekijänoikeuden kaksoissuoja*. Helsinki: Suomalainen lakimiesyhdistys.
- PITKÄNEN, OLLI – LINDFORS, ARTO – PAUNI, SAMI – KELA, SARI – SOININEN, TEEMU

- (2003) Tietotekniikkaoikeus. Teoksessa Hyvönen, Eero (toim.) *Ohjelmistoliiketoiminta*. Helsinki: WSOY, s. 77–118.
- POSKELA, JARNO – KUTINLAHTI, PIRJO – HANHIKE, TIINA – MARTIKAINEN, MIKKO – URJANKANGAS, HANNA-MARIA (2015) *Kokeileva kehittäminen*. Työ- ja elinkeinoministeriön julkaisuja. TEM raportteja. 67/2015. [https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/75056/TEMrap\\_67\\_2015\\_web\\_18112015.pdf?sequence=1](https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/75056/TEMrap_67_2015_web_18112015.pdf?sequence=1).
- PÄÄLLYSAHO, SELIINA – KUUSISTO, JARI (2006) *Osaamisen suojaaminen palveluja kehittämissä yrityksissä*. Helsinki: Tekes.
- RAUTIAINEN, HANNU – ÄIMÄLÄ, MARKUS (2008) *Työsopimuslaki*. Helsinki: WSOY.
- RIEKKINEN, PEKKA (2007) Immateriaalioikeus ja innovaatiopolitiikka: miten mobilisoida luovuus? Teoksessa Mylly, Tuomas – Lavapuro, Juha – Karo, Marko (toim.): *Tekemisen vapaus – Luovuuden ehdot ja tekijänoikeus*. Helsinki: Gaudemus, s. 171–218.
- RIES, ERIC (2016) *Lean Startup. Kokeilukulttuurin käsikirja*. Kerava: LavasDesign.
- ROSEN, LAWRENCE (2004) *Open Source Licensing. Software Freedom and Intellectual Property Law*. Upper Saddle River: Prentice Hall.
- RUFFIN, MICHEL – EBERT, CHRISTOF (2004) Using Open Source Software in Product Development: A Primer. *IEEE Software*, 21(1), 2004, s. 82–86.
- SAARENPÄÄ, AHTI (2001) Teoskynnys, ymmärryskynnys, hyväksymiskynnys. Teoksessa *Asianajotoimisto Borenius & Kemppinen 90 vuotta*. Helsinki: Borenius & Kemppinen, s. 169–197.
- SAMPSON, TIM (2009) Strategic Legal Thinking for IPR Dependent Enterprise – the “Five Rings” Method. *European Intellectual Property Review* 2009, 31(8), s. 411–422.
- SILTALA, RAIMO (2001) *Johdatus oikeusteoriaan*. Helsinki: Helsingin yliopiston oikeustieteellinen tiedekunta.
- SORVARI, KATARIINA (2007) *Tekijänoikeuden loukkaus*. Helsinki: WSOY.
- SPINELLIS, DIOMIDIS – SZYPERSKI, CLEMENS (2004) How Is Open Source Affecting Software Development? *IEEE Software*, 21(1), 2004, s. 28–33.
- TAKKI, PEKKA (2002) *IT-sopimukset – Käytännön käsikirja*. Helsinki: Talentum.

- THOMKE, STEFAN (2001) Enlightened Experimentation – The New Imperative for Innovation. *Harvard Business Review*, February 1/2001, s. 66–75.
- THOMKE, STEFAN (1998) Managing Experimentation in the Design of New Products. *Management Science*, 44(6), s. 743–762.
- TIITINEN, KARI-PEKKA – KRÖGER, TARJA (2012) *Työsopimusoikeus*. Helsinki: Talentum.
- TUULENMÄKI, ANSSI – VÄLIKANGAS, LIISA (2011) The Art of Rapid, Hands-on Execution Innovation. *Strategy & Leadership*, 39(2), s. 28–35.
- TYRVÄINEN, PASI (2003) Ohjelmistojen tuotteistaminen. Teoksessa Hyvönen, Eero (toim.) *Ohjelmistoliiketoiminta*. Helsinki: WSOY, s. 22–40.
- VEDENKANNAS, MATTI (2009) Avoimen innovaatioympäristön oikeudellisesta rakenteesta. *Defensor Legis* 6/2009, s. 961–981.
- VEDENKANNAS, MATTI (2010) Living lab -toimintaympäristön oikeudellisista reunaehdoista. *Defensor Legis* 6/2010, s. 847–868.
- VÄLIMÄKI, MIKKO (2002) Avoimen lähdekoodin ohjelmistolisensseistä. *Defensor Legis* 5/2002, s. 851–861.
- VÄLIMÄKI, MIKKO (2005) *The Rise of Open Source Licensing – A Challenge to the Use of Intellectual Property in the Software Industry*. Helsinki: Turre Publishing.
- VÄLIMÄKI, MIKKO (2009) *Oikeudet tietokoneohjelmistoihin*. Helsinki: Talentum.
- VÄLIMÄKI, MIKKO – LAINE, JUHA (2004) Vastuunrajoituksista kolmannen osapuolen immateriaalioikeusväitteille ohjelmistotoimituksissa. *Defensor Legis* 5/2004, s. 901–911.
- WEST, JOEL – GALLAGHER, SCOTT (2006) Patterns of Open Innovation in Open Source Software. Teoksessa Chesbrough, Henry – Vanhaverbeke, Wim – West, Joel (toim.) *Open Innovation: Researching a New Paradigm*. New York: Oxford University Press, s. 82–106.
- WU, YUHAO – MANABE, YUKI – KANDA, TETSUYA – GERMAN, DANIEL M. – INOUE, KATSURO (2017) Analysis of License Inconsistency in Large Collections of Open Source Projects. *Empirical Software Engineering*, 22(3), June 2017, s. 1194–1222.

## **Virallislähteet**

HE 161/1990 vp.

HE 170/1997 vp.

Bernin yleissopimus kirjallisten ja taiteellisten teosten suojaamisesta. Pariisin sopimuskirja. SopS 79/1986.

Euroopan parlamentin ja neuvoston direktiivi 2001/29/EY tekijänoikeuden ja lähioikeuksien tiettyjen piirteiden yhdenmukaistamisesta tietoyhteiskunnassa

Euroopan parlamentin ja neuvoston direktiivi 2009/24/EY tietokoneohjelmien oikeudellisesta suojasta

KM 1953:5

KM 1987:7

KM 1987:8

Patenttilaki 15.12.1967/550

Perustuslaki 11.6.1999/731

Ratkaisujen Suomi. Pääministeri Juha Sipilän hallituksen strateginen ohjelma 29.5.2015. Hallituksen julkaisusarja 10/2015.

Sopimus teollis- ja tekijänoikeuksien kauppaan liittyvistä näkökohdista (TRIPS-sopimus). SopS 5/1995.

Tekijänoikeuslaki 8.7.1961/404

Työsopimuslaki 26.1.2001/55

WIPO:n tekijänoikeussopimus (WCT). SopS 17/2010.

Yleissopimus eurooppapatenttien myöntämisestä. SopS 8/1996.

## **Oikeuskäytäntö**

EUT C-393/09

EUT C-406/10

EUT C-355/12

I-SHO 7.4.2005, R 03/1060

KKO 1956 II 76

KKO 1984 II 26

KKO 1996:43

KKO 1999:115

KKO 2005:92

KKO 2008:45

## **Tekijänoikeusneuvoston lausunnot**

TN 1996:3

TN 1997:12

TN 2003:10

TN 2003:11

TN 2005:7

TN 2006:5

TN 2006:12

TN 2007:3

TN 2008:13

TN 2011:15

TN 2012:9

TN 2013:1

TN 2013:7

TN 2013:20

TN 2016:15

## Muut lähteet

BALTER, BEN (2015) Open source license usage on GitHub.com. <https://github.com/blog/1964-open-source-license-usage-on-github-com>. Viitattu 25.11.2017.

Black Duck Software: Top Open Source Licenses. <https://www.blackducksoftware.com/top-open-source-licenses>. Viitattu 1.12.2017.

FORTE, STEPHEN: Build an MVP, Not a Beta. <http://fresco.vc/build-an-mvp-not-a-beta>. Viitattu 8.12.2017.

Free Software Foundation: Frequently Asked Questions about the GNU Licenses. <http://www.gnu.org/licenses/gpl-faq.html>. Viitattu 17.11.2017.

Free Software Foundation: The Free Software Foundation (FSF) is a nonprofit with a worldwide mission to promote computer user freedom. <http://www.fsf.org/about>. Viitattu 24.11.2017.

Free Software Foundation: What Is Free Software? <https://www.gnu.org/philosophy/free-sw.html>. Viitattu 24.11.2017.

KÄLVIÄINEN, MIRJA (2016) Kokeilukulttuuri muotoiluprosessin tekemisen tapana. *UAS Journal* 3/2016. <https://uasjournal.fi/puheenvuoro/kokeilukulttuuri-muotoiluprosessin-tekemisen-tapana>. Viitattu 11.4.2017.

MANNER, MIKKO (2001) Avoin lähdekoodi ohjelmistotaloudessa. *IPRinfo* 3/2001. [https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2001/IPRinfo\\_3-2001/fi\\_FI/Avoin\\_lahdekoodi\\_ohjelmistotaloudessa](https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2001/IPRinfo_3-2001/fi_FI/Avoin_lahdekoodi_ohjelmistotaloudessa). Viitattu 24.4.2017.

NICKOLS, FRED (2016) *Strategy: Definitions and Meaning*. [http://www.nickols.us/strategy\\_definitions.pdf](http://www.nickols.us/strategy_definitions.pdf). Viitattu 16.5.2017.

OKSANEN, MARKUS (2004) Ohjelmakoodi tekijänoikeussuojan kohteena. *IPRinfo* 3/2004.

[https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2004/IPRinfo\\_3-2004/fi\\_FI/Ohjelmakoodi\\_tekijanoikeussuojan\\_kohteena](https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2004/IPRinfo_3-2004/fi_FI/Ohjelmakoodi_tekijanoikeussuojan_kohteena). Viitattu 24.4.2017.

Open Source Initiative: About the Open Source Initiative. <https://opensource.org/about>. Viitattu 24.11.2017.

Open Source Initiative: GNU General Public License version 2. <https://opensource.org/licenses/GPL-2.0>. Viitattu 24.11.2017.

Open Source Initiative: Open Source Definition. <https://opensource.org/osd>. Viitattu 24.11.2017.

Open Source Initiative: The MIT License. <https://opensource.org/licenses/MIT>. Viitattu 25.11.2017.

PYKÄLÄ, TIMO (2008) Ohjelmistopatentit tarpeen innovaatioiden suojaksi – Tekijänoikeus riittää vain suoraa kopiointia vastaan. *IPRinfo* 4/2008. [https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2008/IPRinfo\\_4-2008/fi\\_FI/Ohjelmistopatentit\\_tarpeen\\_innovaatioiden\\_suojaksi\\_\\_Tekijanoikeus\\_riittaa\\_vain\\_suoraa\\_kopiointia\\_vastaan](https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2008/IPRinfo_4-2008/fi_FI/Ohjelmistopatentit_tarpeen_innovaatioiden_suojaksi__Tekijanoikeus_riittaa_vain_suoraa_kopiointia_vastaan). Viitattu 24.4.2017.

ROSEN, LAWRENCE (2001) The Unreasonable Fear of Infection. <http://www.rosenlaw.com/html/GPL.pdf>. Viitattu 30.11.2017.

SANDVIK, RISTO (2011) Ketterä ohjelmistokehitys ja toimitussopimukset. *IPRinfo* 3/2011. [https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2011/IPRinfo\\_3-2011/fi\\_FI/Kettera\\_ohjelmistokehitys\\_ja\\_toimitussopimukset](https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2011/IPRinfo_3-2011/fi_FI/Kettera_ohjelmistokehitys_ja_toimitussopimukset). Viitattu 24.4.2017.

VON WILLEBRAND, MARTIN (2012) Protecting the Business of a Gaming Company. *IPRinfo* 2/2012. [https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2012/IPRinfo\\_2-2012\\_eng/fi\\_FI/Protecting\\_the\\_Business\\_of\\_a\\_Gaming\\_Company](https://www.iprinfo.com/julkaisut/iprinfo-lehti/lehtiarkisto/2012/IPRinfo_2-2012_eng/fi_FI/Protecting_the_Business_of_a_Gaming_Company). Viitattu 24.4.2017.

World Intellectual Property Organization: WIPO-Administered Treaties. [http://www.wipo.int/treaties/en/ShowResults.jsp?lang=en&treaty\\_id=15](http://www.wipo.int/treaties/en/ShowResults.jsp?lang=en&treaty_id=15). Viitattu 22.12.2017.

## LYHENTEET

Art.	Artikla
BSD	Berkeley Software Distribution
Euroopan patenttisopimus	Yleissopimus eurooppapatenttien myöntämisestä, SopS 8/1996.
EUT	Euroopan unionin tuomioistuin
FSF	Free Software Foundation
GNU	GNU's Not Unix
GPL	GNU General Public License
HE	Hallituksen esitys
IPR	Intellectual Property Rights
I-SHO	Itä-Suomen hovioikeus
KKO	Korkein oikeus
KM	Komiteanmietintö
LGPL	GNU Lesser General Public License
MIT	Massachusetts Institute of Technology
Ohjelmistodirektiivi	Euroopan parlamentin ja neuvoston direktiivi 2009/24/EY tietokoneohjelmien oikeudellisesta suojasta
OSD	Open Source Definition
OSI	Open Source Initiative
OSS	Open Source Software
PatL	Patenttilaki 15.12.1967/550
PL	Perustuslaki 11.6.1999/731
TekijäL	Tekijänoikeuslaki 8.7.1961/404
Tietoyhteiskuntadirektiivi	Euroopan parlamentin ja neuvoston direktiivi 2001/29/EY tekijänoikeuden ja lähioikeuksien tiettyjen piirteiden yhdenmukaistamisesta tietoyhteiskunnassa
TN	Tekijänoikeusneuvosto
TRIPS-sopimus	Sopimus teollis- ja tekijänoikeuksien kauppaan liittyvistä näkökohdista (Agreement on Trade-Related Aspects of Intellectual Property Rights), SopS 5/1995
TSL	Työsopimuslaki 26.1.2001/55
WIPO	World Intellectual Property Organization (Maailman henkisen omaisuuden järjestö)



# 1. JOHDANTO

## 1.1 Johdatus aiheeseen

Tietoyhteiskunnassa yhä suurempi osa – yleisen arvion mukaan noin 75 prosenttia – yrityksen varallisuudesta koostuu aineettomasta omaisuudesta, joka usein on keskeinen tekijä myös yrityksen kilpailuedun kannalta<sup>1</sup>. Aineetonta varallisuutta suojaavien yksinoikeuksien eli immateriaalioikeuksien merkitys liiketoiminnassa on digitalisaation myötä kasvanut voimakkaasti, sillä nopea ja edullinen digitaalinen kopiointi on mahdollistanut irrottautumisen perinteisten kopiointimetodien asettamista rajoituksista. Immateriaalioikeuksista tekee liiketoiminnan kannalta houkuttelevia myös niiden erikoispiirre fyysiseen omaisuuteen nähden: ne eivät vähene tai kulu käytössä, vaan ne voivat tuottaa yritykselle jatkuvia tuloja.<sup>2</sup> Kuitenkaan näillä oikeuksilla sinänsä ei ole itseisarvoa, vaan niitä on osattava hyödyntää kannattavalla tavalla<sup>3</sup>. Myös aineettomaan varallisuuteen liittyvät riskit ovat nykyään entistä suurempia<sup>4</sup>. Jotta yritys voisi saada immateriaalioikeuksistaan suurimman mahdollisen hyödyn ja välttää muiden immateriaalioikeuksien loukkaamisesta seuraavat vastuut, tulee sen ottaa aineettoman varallisuuden hallinta osaksi strategiaansa.

Perinteisesti immateriaalioikeuksien laajamittainen taloudellinen hyödyntäminen on liitetty lähinnä teollisoikeuksiin, mutta tekijänoikeuksien teollista painoarvoa on lisännyt erityisesti tietokoneohjelmien ja tietokantojen kuuluminen tekijänoikeudellisen suojan piiriin<sup>5</sup>. Yksi voimakkaimmin viime vuosina kasvaneista tekijänoikeuden ydinaloista onkin juuri ohjelmistoteollisuus<sup>6</sup>. Ohjelmistoala on tekijänoikeudellisesta näkökulmasta kiinnostava kohde taloudellisen merkittävyytensä vuoksi, mutta myös siksi, että tietokoneohjelmat ovat tekijänoikeutta ravistelevan digitalisaation ja tietoyhteiskuntakehityksen keskiössä. Oikeustieteen piirissä onkin käyty runsaasti keskustelua tietokoneohjelmien immateriaalioikeudellisesta luonteesta, eikä nykyistä ratkaisua ohjelmien tekijänoikeudellisesta suojaamisesta ole hyväksytty kritiikittä.

Teokset, kuten tietokoneohjelmat, syntyvät luovan työn tuloksena. Vaikka teosten tuottaminen voidaan romantisoidusti kuvitella yksittäisen taiteilijaneron itseilmaisun muodoksi, liike-elämässä luova toiminta (esimerkiksi tuotekehitys) tapahtuu todennäköisemmin

---

1 Innanen – Jäske 2014, s. 16 ja Luoma – Huuskonen 2008, s. 29

2 Innanen – Jäske 2014, s. 23

3 Haarmann – Mansala 2012, s. 183

4 Palfrey 2012, s. 9

5 Lahti 2006, s. 109

6 Grönlund ym. 2014, s. 4–5

ryhmytyönä järjestelmällisesti toteutetussa prosessissa. Viime vuosina tällaisessa innovointityössä on alettu painottaa ns. ketteriä menetelmiä, kuten kokeillen kehittämistä. Kun pitkällisen suunnittelun sijasta luovaan työhön ryhdytään nopeilla ja edullisilla kokeiluilla, saadaan kehitettävästä ideasta testattua tietoa jo prosessin alkuvaiheessa. Idean toimivuudesta kokeilemalla saatu tieto puolestaan auttaa suuntaamaan kehitystyössä tarvittavia resursseja oikeisiin asioihin.<sup>7</sup> Kokeilujen kautta pystytään moniaistillisesti osallistamaan prosessiin kehitettävänä olevan tuotteen käyttäjiä, jolloin aidot käyttäjätarpeet voidaan huomioida jo varhain – ja siten suunnitella tuotteita, joille on todellisuudessa kysyntää<sup>8</sup>. Kokeilevan toimintatavan uskotaan luovan innovaatiotoimintaan motivoivaa ilmapiiriä, edistävän yhteistä ajattelua ja lisäävän kustannustehokkuutta<sup>9</sup>. Kuitenkin ongelmalliseksi on koettu kokeilukulttuurin juurruttaminen sekä kokeilemista kangistava lainsäädäntö.<sup>10</sup>

Tutkielmassani risteävät kolme edellä kuvattua ajankohtaista teemaa: tarve immateriaalioikeuksien strategiseen hallintaan, digitalisaation tuomien uusien teoslajien ja niiden käyttötapojen tekijänoikeudellinen sääntely sekä ketterien menetelmien omaksuminen innovaatiotoiminnassa. Tutkielma käsittelee sitä, millaisiin tekijänoikeudellisiin seikkoihin ohjelmistoalan yrityksen tulisi immateriaalioikeusstrategiassaan kiinnittää huomiota, kun se hyödyntää tuotekehityksessään kokeiluja. Pyrin erityisesti pureutumaan tiettyihin kokeilujen ominaispiirteisiin ja muodostamaan niihin liittyviä toimintaehdotuksia koskien sitä, miten yritys voi strategisesti hyötyä tekijänoikeuksista ja välttää pahimmat karikot.

Edellä viitatus taloudellisen merkityksensä lisäksi tekijänoikeudella on katsottu olevan tärkeä rooli suhteessa sosiaaliseen ja kulttuuriseen kehitykseen, sillä sen on nähty kannustavan yksilöitä luovaan toimintaan ja tuottavan siten tekijän saaman hyödyn ohella myös yhteistä hyvää. Kun tekijänoikeutta tarkastellaan sen oletetun kannustinvaikutuksen näkökulmasta, vaikuttavat tekijänoikeus ja kokeilukulttuuri ilmentävän samansuuntaisia päämääriä. Sekä tekijänoikeussuojalla että kokeiluilla pyritään rohkaisemaan yksilöitä luovaan toimintaan, ja innovoinnin vilkastumisen toivotaan vaikuttavan yhteiskunnalliseen kehitykseen laajemminkin. Toisaalta tekijänoikeudella on yksinoikeusluonteensa vuoksi myös luovaa toimintaa rajoittava vaikutus. Vaikka keskitynkin tekstissäni käytännönläheisempiin ohjelmistoalan yrityksen tekijänoikeudellisia toimia koskeviin kysymyksiin, kytkeytyy tutkielmani pohjimmiltaan keskusteluun tekijänoikeuden ja luovuuden suhteesta.

7 Hassi – Tuulenmäki 2012, s. 2

8 Kälviäinen 2016

9 Kälviäinen 2016

10 Berg 2013, s. 18, 22 ja Berg ym. 2014, s. 25

## 1.2 Tutkimuskysymykset ja raja

Kokeillen kehittämisen keskeisiä tavoitteita ovat nopeus ja kustannustehokkuus. Idean kehittelyn ja kokeilun suunnittelun pitäisi sujua rivakasti, eikä kokeilussa käytettävän prototyypin luomiseen saisi kulua kovin paljon aikaa tai rahaa. Kokeiluasetelman valmisteluvaiheessa on ratkaistava, tuotetaanko kaikki prototyyppiin sisältyvä tekijänoikeudellinen materiaali itse vai pyritäänkö resursseja säästämään käyttämällä suunnittelutyön pohjana valmista, jonkun toisen tuottamaa materiaalia. Tarkoitukseni on tutkia, millaisin tavoin yritys voi laillisesti hyödyntää ohjelmistokehitykseen pyrkivässä kokeilussa ulkopuolisten tekijänoikeudellista materiaalia siten, että kokeilu voidaan viedä läpi mahdollisimman ”keveästi” (luku 5).

Kokeileminen on ideaa kehittävän kehitystiimin ja kokeiluun osallistuvien koehenkilöiden tai oikeiden asiakkaiden välistä vuorovaikutusta. Kun teoksen luomiseen osallistuu useita henkilöitä, ei välttämättä ole selvää, kenelle tekijänoikeus syntyy. Kun luovaa työtä toteutetaan liiketoiminnan yhteydessä, on yrityksen usein kannattavaa pyrkiä siirtämään syntyvät oikeudet itselleen. Erityisesti luvussa 6 pyrin hahmottamaan kokeiluun osallistuvien henkilötahojen välisiä suhteita tekijänoikeudellisesta näkökulmasta ja löytämään toimintatapoja, joiden avulla kokeilun suorittava yritys voi huolehtia kokeilussa syntyvien tekijänoikeuksien siirtymisestä itselleen.

Ohjelmistoalan yrityksen IPR-strategian kokonaisvaltaisen käsittelyn sijasta nostan tutkielmassani esiin sellaisia strategian elementtejä, jotka ovat relevantteja juuri kokeilevan innovaatiotoiminnan kannalta. Koska kaikkien immateriaalioikeudellisten suojamuotojen tarkasteleminen ei tutkielman laajuuden puitteissa ole mahdollista, olen rajannut esitykseni koskemaan lähinnä tekijänoikeudellisia näkökohtia. Ohjelmistokehityksen kohdalla tämä on perusteltua myös siksi, että tekijänoikeus on tietokoneohjelmien pääasiallinen suojamuoto<sup>11</sup>. Seitsemännessä luvussa selvitän kuitenkin tiiviisti, millainen rooli tekijänoikeudella on kokeilevan ohjelmistoyrityksen IPR-strategiassa suhteessa muihin alalla yleisesti käytettyihin suojakeinoihin.

Edellä kuvailtujen alakysymysten kautta pyrin lopulta luomaan kokonaiskuvan siitä, millaisia tekijänoikeudellisia kysymyksiä ohjelmistokehitykseen tähtäävän kokeilun suorittamiseen liittyy ja millaisin strategisoin valinnoin tekijänoikeuksia voidaan ko. kysymyksissä hallita kannattavasti.

<sup>11</sup> Luoma – Huuskonen 2008, s. 34

### 1.3 Metodi

Käytän tutkielmassani lainopillista eli oikeusdogmaattista metodologiaa. Lainopillinen tutkimus systematisoi ja tulkitsee voimassa olevan oikeuden oikeussääntöjä, ja tulkinnat perustellaan vallitsevan oikeuslähteopin mukaisilla lähteillä<sup>12</sup>. Tässä tutkielmassa oikeuslähteinä käytetään lainsäädäntöä, lain esitöitä, oikeuskirjallisuutta, tuomioistuinratkaisuja sekä tekijänoikeusneuvoston lausuntoja. Avointa lähdekoodia koskien huomioidaan myös Open Source -ohjelmistokehityksen yhteisönormit<sup>13</sup>. Erityisen keskeistä lainopillisen tulkinnan muodostamisessa on lain kielellis-semanticisten merkitysten, esitöiden ilmentämän lainsäätäjän tarkoituksen sekä oikeuskäytännössä vakiintuneiden merkitysten analysoiminen<sup>14</sup>. Vaikka tutkielmani tavoitteet eivät ole puhtaasti juridisia, vaan ne liittyvät kiinteästi liiketaloudellisiin hyötynäkökohtiin, vaatii yrityksen immateriaalioikeusstrategian toimintavaihtoehtojen hahmottelu sen selvittämistä, miten voimassa oleva oikeus suhtautuu ohjelmistojen kehittämiseen kokeilujen avulla.

Oikeusdogmaattista otetta immateriaalioikeudellisessa tutkimuksessa on kritisoitu siitä, että se ei riittävästi kyseenalaista immateriaalioikeudellisen lainsäädännön taustalla olevia poliittisia ja ideologisia näkökohtia, vaan oikeuslähteisiin nojaava argumentaatio oletetaan järkeväksi ja epäpoliittiseksi<sup>15</sup>. Tutkielmani tavoitteena ei kuitenkaan ole tekijän yksinoikeusaseman oikeutuksen arviointi tai tekijänoikeuden laajempien yhteiskunnallisten vaikutusten kartoittaminen, vaan tarkastelen tutkimuskysymyksiäni yksittäisen yrityksen näkökulmasta. Vaikka tietyt tekijänoikeudellisen sääntelyn piirteet voidaan nähdä innovaatiotoimintaa jarruttavina tekijöinä, tavoitteeni ei niinkään ole löytää lainsäädäntötoimien tai lainsäädännön purkamisen kohteita vaan pikemminkin hahmottaa kannattavia toimintatapoja yritykselle, jonka pyrkimyksenä on menestyä nykyisessä toimintaympäristössä voimassa olevan lainsäädännön asettamissa kehyksissä. Näin ollen näkökulmani on enemmän vallitsevaan kuin kriittiseen lainoppiin tukeutuva<sup>16</sup>. Kuitenkin tutkielmassani nousee esiin myös tekijänoikeuslainsäädännön luovuutta edistäviä tai estäviä piirteitä koskevia huomioita.

12 Siltala 2001, s. 17

13 Välimäen 2002, s. 854, mukaan yhteisönormit on huomioitava sekundäärisinä lähteinä, sillä ne heijastelevat lisenssisopimuksen tarkoitusta. Tällaisia yhteisönormeja ovat mm. yhteisön eettiset ohjeet ja usein kysyttyihin kysymyksiin annetut vastaukset.

14 Siltala 2001, s. 122

15 Ks. Riekkinen 2007, s. 174–179

16 Siltalan 2001, s. 18–19, mukaan lainoppi voidaan jakaa vallitsevaan lainoppiin sekä vaihtoehtoiseen, kriittiseen tai poliittiseen lainoppiin. Vallitseva lainoppi tukeutuu sellaisiin lähteisiin ja tulkintatapoihin, jotka ovat yleisesti hyväksytyjä myös käytännön oikeudellisessa ratkaisutoiminnassa, ja se pyrkii tuottamaan ylimpien tuomioistuinten ratkaisukäytännön kanssa yhdenmukaisia tulkintoja. Lainopin kriittinen puoli taas irtautuu tuomarimaisesta oikeuden tulkintatavasta ja pyrkii perustelemaan oikeuden parhaita tulkintatapoja esimerkiksi jonkin ideologian, tasa-arvon tai taloustieteen näkökulmasta.

## 1.4 Tutkielman rakenne

Johdantoa seuraava luku käsittelee kokeiluja osana yrityksen innovaatio- ja tuotekehitystoimintaa, koska kokeilujen ymmärtäminen menetelmänä on keskeistä niiden oikeudellisen analyysin mahdollistamiseksi. Luvussa kuvaan lyhyesti kokeiluilla tavoiteltuja hyötyjä ja menetelmän suhdetta perinteiseen suunnitteluun, mutta keskityn erityisesti kokeilusyklin eri vaiheisiin. Kolmannessa luvussa selvitän, mitä IPR-strategialla tarkoitetaan ja millaiset valinnat tyypillisesti kuuluvat IPR-strategian piiriin.

Neljäs luku kokoaa tutkielman tekijänoikeudelliset lähtökohdat. Luvussa jäsennän ja tulkiten teoksen käsitettä ja tekijän yksinoikeuksia koskevia tekijänoikeuslain säännöksiä tietokoneohjelmien ja kokeilemisen kannalta. Luvun tarkoituksena on hahmottaa tietokoneohjelman prototyypin testaamiseksi luotu kokeiluasetelma tekijänoikeudellisesta näkökulmasta.

Viidennessä luvussa tutkin, millainen tekijänoikeuksien käyttäminen mahdollistaa kokeilun läpiviennin mahdollisimman vähällä vaivalla, ajalla ja rahalla. Tarkastelun keskiöön nousee valmiiden komponenttien lisensoiminen ja erityisesti avoimen lähdekoodin hyödyntäminen osana prototyyppiä. Kuudennessa luvussa tarkastelen kokeilun henkilöllistä ulottuvuutta ja pyrin selvittämään, kuka tai ketkä kokeilussa voivat olla teoksen tekijöitä ja miten kokeilun toteuttava yritys voi päästä kokeilussa syntyvien tekijänoikeuksien haltijaksi.

Seitsemännen luvun tarkoitus on luoda kokeilevan ohjelmistoyrityksen IPR-strategiasta tiiviisti hieman laajempi kuva, jossa nostetaan esiin myös ohjelmistopatentin ja ei-muodollisten suojauskeinojen kannalta olennaisia huomioita. Lisäksi arvioin tekijänoikeutta suhteessa em. keinoihin sekä avointa innovaatiota yrityksen strategiana. Viimeinen luku kiteyttää tutkielman johtopäätökset ja tutkimuskysymyksiin saadut vastaukset.

## 2. KOKEILUT INNOVAATIOPROSESSISSA

### 2.1 Kokeilut, kokeilukulttuuri ja kokeiluyhteiskunta

Muotoilu- ja tuotekehitysprosesseissa viimeistelemättömillä prototyypeillä suoritettavilla kokeiluilla on ollut merkittävä rooli jo pitkään, mutta viime vuosina kokeilutoiminnan mahdollisuudet on tunnistettu laajemminkin<sup>17</sup>. Suomessa kokeilevia toimintatapoja pyritään parhaillaan omaksumaan niin julkisella kuin yksityiselläkin sektorilla<sup>18</sup>, ja kokeilukulttuurin käyttöön ottaminen on kirjattu 2015 aloittaneen hallituksen strategiaan tavoitteisiin<sup>19</sup>. Myös teknologisen kehityksen suhteen aika vaikuttaisi olevan kypsä nopeille ja kehämäisesti toteutettaville kokeiluille, jotka ovat nyt helpompia ja halvempia toteuttaa kuin koskaan<sup>20</sup>.

Kokeileminen on osa ns. ketterää kehittämistä (*agile development*), jossa suositetaan kevyttä suunnittelua, ripeitä kehityssyklejä sekä jouhevaa vuorovaikutusta ankaran suunnitelmallisuuden sijasta<sup>21</sup>. Kokeilu on empiirinen testi ja iteratiivinen oppimissykli, jossa yritys ja erehdys johdattavat kehitystiimiä kohti käsillä olevan suunnitteluongelman parempaa ymmärtämistä<sup>22</sup>. Kevein kokeiluin kehitettävästä ideasta saadaan palautetta nopeasti, ja kokeilemalla saadun tiedon avulla voidaan vahvistaa tai kyseenalaistaa ideaan liittyviä oletuksia<sup>23</sup>. Menetelmällä pystytään siis välttämään resurssien haaskausta huonoihin ideoihin, kun hypoteesien toimivuus voidaan jo alkumetreillä todentaa käytännössä. Kun kokeilut toteutetaan mahdollisimman pieninä ja yksinkertaisina, eivät riskitkään kasva suuriksi<sup>24</sup>.

Kokeiluiden voima perustuu siihen, että kokeilemalla saadaan informaatiota, jota intuitio ei voi osoittaa<sup>25</sup>. Päätöksiä ei siten tarvitse perustaa pelkille oletamuksille, vaan kokeiluiden synnyttämän tiedon varassa prosessia voidaan ohjata oikeaan suuntaan. Erityisen soveltuvia kokeilut ovat tavoiteltaessa innovaatioita, jotka ovat aidosti uusia ja joiden kehittämiseen perinteisen projektinhallinnan keinot soveltuvat huonosti, koska suunnitelmien tekemiseen tarvittavaa tietoa ei ole saatavilla<sup>26</sup>. Tällaisille innovaatioprosesseille tyypillistä on tulosten heikko ennustettavuus sekä kausaalisuhteiden monimutkaisuus, minkä

17 Hassi – Tuulenmäki 2012, s. 2

18 Berg 2013, s. 5

19 Ratkaisujen Suomi. Pääministeri Juha Sipilän hallituksen strateginen ohjelma. 29.5.2015. Hallituksen julkaisusarja 10/2015, s. 27

20 Thomke 2001, s. 68. Thomke mainitsee kokeilutoimintaa helpottaneina uutuuksina esim. tietokonesimulaatiot ja nopean prototypoinnin.

21 Berg – Hildén – Lahti 2014, s. 16–17

22 Hassi – Tuulenmäki 2012, s. 7, 9

23 Berg ym. 2014, s. 5 ja Thomke 2001, s. 69

24 Hassi – Paju – Maila 2015, s. 28

25 Davenport 2009, s. 70

26 Hassi ym. 2015, s. 9 ja Tuulenmäki – Välikangas 2011, s. 28

vuoksi ratkaisujen luominen tulevaisuuteen suuntautuvan kokeilutoiminnan kautta on helpompaa kuin olemassa olevan tiedon varaan suunnitellen<sup>27</sup>.

Kokeilevien toimintamallien omaksuminen edellyttää sosiaalista ympäristöä, jossa kokeiluihin kannustetaan ja epäonnistumisia ei tuomita – puhutaan kokeilukulttuurista. Kokeiluyhteiskunnaksi voidaan puolestaan kutsua yhteiskuntaa, jossa vallitsee kokeilukulttuuri ja jossa kokeiluihin rohkaistaan kannustimin sekä kokeilujen esteitä poistaen.<sup>28</sup> Toimiva kokeiluympäristö ei siis edellytä vain tiettyjä teknologisia ja strategisia valmiuksia, vaan myös oikeaa ilmapiiriä. Raskaaseen suunnitteluun tottuneissa organisaatioissa täytyy ennen menestyksekkäitä kokeiluita kehittää erehdyksiä sietävä kulttuuri<sup>29</sup>.

## 2.2 Suunnitteleminen ja kokeileminen

Perinteisesti innovaatioprosessit ovat perustuneet tarkalle ennakkosuunnittelulle<sup>30</sup>. Suunnitteluajattelun mukaan ongelmat voidaan ratkaista laatimalla suunnitelma ja noudattamalla sitä täsmällisesti, sillä prosessin ihanteellisen suunnan ajatellaan olevan nähtävissä jo työskentelyn varhaisessa vaiheessa<sup>31</sup>. Suunnitteluun perustuvassa projektissa idean kehitys nähdään tiettyä päämäärää kohti etenevänä lineaarisena, loogis-systemaattisena jatkumona, kun taas kokeillen kehittäminen on elävä prosessi vailla ennalta määrättyä lopputulosta<sup>32</sup>.

Suunnitelma on kuvaus siitä, mitä tulee tapahtumaan<sup>33</sup>. Näin ollen suunnittelu soveltuu vain tilanteisiin, joissa on ennalta mahdollista kerätä riittävästi tietoa suunnitelman laatimiseksi<sup>34</sup>. Tällaista vaadittavaa tietoa ovat yksityiskohdat koskien sitä, mitä ollaan tekemässä, millaisia vaiheita projektiin sisältyy ja millaisia seurauksia toimilla tulee olemaan. Myös epävarmuustekijät tulee olla ennakoitavissa niin hyvin, että ne voidaan huomioida suunnitelmassa. Suunnittelun voidaankin karkeasti todeta soveltuvan tilanteisiin, joissa halutaan aikaansaada tuttuja tuloksia tutuissa ympäristöissä mullistavien innovaatioiden sijasta.<sup>35</sup> Kokeileminen puolestaan on suunnittelemista toimivampi lähestymistapa silloin, kun suunnittelu on mahdotonta saatavilla olevan tiedon puutteen vuoksi. Tällöin tieto on

27 Hassi – Tuulenmäki 2012, s. 2

28 Berg ym. 2014, s. 14

29 Berg 2013, s. 18

30 Hassi – Tuulenmäki 2012, s. 5

31 Berg ym. 2014, s. 15 ja Kälviäinen 2016

32 Hassi ym. 2015, s. 25, Kälviäinen 2016 ja Tuulenmäki – Välikangas 2011, s. 31

33 Hassi ym. 2015, s. 19

34 Hassi – Tuulenmäki 2012, s. 5

35 Hassi ym. 2015, s. 20–21

itse tuotettava kokeilemalla ideaa käytännössä.<sup>36</sup> Uusiin innovaatioihin liittyy aina tällaista tiedollista epävarmuutta, koska epävarmuutta vailla olevassa prosessissa myös toimien lopputulos on ennalta tunnettu. Kokeilemalla kehitettäessä projektin lopputulos alkaa hahmottua vasta iteratiivisten kokeilusyklien toiston myötä, kun kokeilut tuottavat uutta tietoa, jota hyödynnetään seuraavissa kokeiluissa.<sup>37</sup> Uutuusarvoon liittyvän epävarmuuden lievittämisen ohella kokeileminen sopii hyvin monimutkaisiin tilanteisiin, joissa asioiden välisten syy-seuraussuhteiden hahmottaminen ennalta on hankalaa<sup>38</sup>.

Siinä missä suunnitteleminen mahdollistaa pitkällisen tukeutumisen pelkkiin mielikuviin, vaatii kokeiluihin turvautuminen käytännön asiakaskontaktointia jo varhaisessa vaiheessa. Sen sijaan, että prosessi toteutettaisiin suljetusti vain luovan tiimin kesken, kokeiluihin osallistetaan koehenkilöitä tai oikeita asiakkaita.<sup>39</sup> Suunnitteluun tukeutuvissa projekteissa luova toiminta ja käytännön toteutus on erotettu omiksi toiminnoikseen, mutta kokeiluissa luova ideointi tapahtuu nimenomaan tekemisen kautta<sup>40</sup>. Idean muovaaminen kokeiluksi pakottaa kokeiluun osallistuvat konkretiaan ja asioiden täsmällisempään määrittelyyn pelkkien tulkinnanvaraisten sanojen sijasta. Samalla idean tuominen fyysiseen muotoon auttaa myös havaitsemaan sen aukot ja ongelmakohdat.<sup>41</sup>

## 2.3 Kokeilu syklinä

### 2.3.1 Kokeilun vaiheet

Kokeilua voidaan kuvata iteratiivisena kiertona, joka koostuu neljästä vaiheesta: 1) kokeilun suunnitteleminen, 2) kokeiluaselman valmisteleminen, 3) kokeilun suorittaminen ja 4) tulosten analysoiminen.<sup>42</sup> Viimeinen vaihe, tulosten analysoiminen ja opitun reflektointi, luo pohjan seuraavalle kokeilusyklille<sup>43</sup>. Oppimista tapahtuu vaihe vaiheelta ja kierros toisensa jälkeen, ja jokaisen iteraation päätteeksi ideaa on mahdollista parannella kokeilulla kerätyn tiedon mukaan<sup>44</sup>. Seuraavissa alaluvuissa kokeiluprosessi esitetään tämän neliportaisen mallin mukaisesti.

36 Hassi – Tuulenmäki 2012, s. 5

37 Hassi ym. 2015, s. 24–25

38 Hassi ym. 2015, s. 21

39 Hassi ym. 2015, s. 4

40 Tuulenmäki – Välikangas 2011, s. 32

41 Kälviäinen 2016

42 Thomke 1998, s. 744 ja Hassi ym. 2015, s. 39, hahmottavat syklin vaiheet muuten samankaltaisesti, mutta Hassi ym. erottavat kokeilun suunnittelua edeltäväksi erilliseksi vaiheeksi idean osien epävarmuuksien tunnistamisen. Riesin 2016, s. 96, laatima rakenna-mittaa-opi-palautesykli on hieman pelkistetympi: se koostuu ns. pienimmän toimivan tuotteen rakentamisesta, tuotteen testaamisesta kokeilulla ja kokeilusta saadun tiedon pohjalta tapahtuvasta oppimisesta.

43 Hassi ym. 2015, s. 39

44 Thomke 1998, s. 744 ja Hassi – Tuulenmäki 2012, s. 8



Hassin ja Tuulenmäen mukaan kokeilusyklin tavoitteena on tunnistaa, luoda ja kerätä kolmentyyppisiä ideoita: kehitettäviä ideoita (*opportunity ideas*), kokeilun ideoita (*experimentation ideas*) ja toteuttamisen ideoita (*execution ideas*)<sup>45</sup>. Kehitettävät ideat ovat vaihtoehtoisia ratkaisuja asiakastarpeeseen, kun taas kokeilun ideat ovat mahdollisia tapoja kokeilla kehitettävää ideaa. Toteuttamisen ideat puolestaan ovat kokeilemalla vahvistettuja oppeja siitä, miten kehitettävä idea voitaisiin käytännössä toteuttaa. Kokeilemalla pyritään luomaan uusia toteuttamisen ideoita sekä tarkentamaan aiempia ideoita.<sup>46</sup>

### 2.3.2 Kokeilun suunnitleminen

Vaikka kokeileminen voidaan nähdä perinteiselle suunnittelulle vaihtoehtoisena kehittämistapana, vaatii kokeilun toteuttaminenkin alussa suunnittelua. Kokeilusuunnitelma, joka määrittää kokeilun vaiheet, tavoitteet ja siinä tarvittavat valmistelut, tekee prosessista sekä teknisesti että psykologisesti helpomman.<sup>47</sup> Kokeilun suunnittelun lähtökohtana on kehitettävä idea, joka sisältää tunnistetun ongelman ja mahdollisen ratkaisun kyseiseen ongelmaan<sup>48</sup>. Kokeilu suunnitellaan empiirisen kokeen malliin siten, että testattava ratkaisu sisältää hypoteesin, jonka paikkansapitävyyttä kokeilu mittaa<sup>49</sup>. Koska kokeiluiden on tarkoitus tuottaa tietoa nopeasti, on prosessi syytä jaksottaa enintään muutaman viikon mittaisiin etappeihin, joiden päätteeksi on ratkaistava kehitettävään ideaan kohdistettavat jatkotoimenpiteet<sup>50</sup>. Kokeilun suunnitleminen on yhtä lailla luovaa toimintaa kuin kehitettävän idean keksiminenkin, sillä kokeilemisen tapoja on lukuisia ja yhtä parasta tapaa ei ole. Suunnitteluvaiheessa on ratkaistava, millaisia prototyyppisiä ja muita valmisteluja kokeilu vaatii ja miten kokeilemalla saatava tieto kerätään.<sup>51</sup>

### 2.3.3 Kokeiluasetelman valmistelu

Kokeiluasetelma on tilanne, jossa koehenkilö on vuorovaikutuksessa kokeilussa käytettävän prototyypin kanssa<sup>52</sup>. Hassi ym. tuovat esille kuusi erilaista vaihtoehtoa kokeiluasetelman luomiseen:

- **SANALLINEN SELITYS.** Kehitettävästä ideasta kerrotaan koehenkilölle sanallisesti. Verbaalisen kuvauksen tukena käytetään prototyyppiä tai muuta konkretisointia.
- **LÄPIKÄYNTI.** Koehenkilölle osoitetaan tietty rooli sekä tehtävä, joka hänen on suoritettava.

45 Hassi – Tuulenmäki 2012, s. 8. Termien suomennotukset teoksesta Hassi ym. 2015, s. 42.

46 Hassi ym. 2015, s. 43–44

47 Hassi ym. 2015, s. 78

48 Hassi – Tuulenmäki 2012, s. 9

49 Davenport 2009, s. 72 ja Ries 2016, s. 75

50 Hassi ym. 2015, s. 69

51 Hassi – Tuulenmäki 2012, s. 9–10

52 Hassi ym. 2015, s. 100

tava prototyyppiä käyttämällä.

- **ROOLIPELI.** Tilanne näytellään ennalta laaditun käsikirjoituksen mukaisesti todellisen käyttötilanteen ulkopuolella.
- **SIMULAATIO.** Asetelma rakennetaan mahdollisimman paljon oikeaa käyttötilannetta muistuttavaksi, mutta esim. automatisoituja toimintoja voidaan toteuttaa manuaalisesti.
- **SAVUTESTI.** Asiakkaille annetaan mahdollisuus jättää ennakkotilaus valmiilta vaikuttavasta tuotteesta tai palvelusta, jota ei todellisuudessa ole vielä olemassa.
- **OZIN VELHO.** Asiakkaalle annetaan mahdollisuus käyttää palvelua, jota hän luulee automatisoiduksi, mutta joka vielä kokeiluvaiheessa toteutetaan manuaalisesti.<sup>53</sup>

Prototyyppiä kutsutaan kokeilussa käytettävää oppimisen apuvälinettä, jonka avulla koehenkilö saa aidomman kokemuksen ja kehitystiimi paremman ymmärryksen kehitettävästä ideasta<sup>54</sup>. Prototyyppi voi olla esimerkiksi kaksiulotteinen visualisointi, kolmiulotteinen esine, digitaalinen prototyyppi tai jopa tila (esim. lavastettu tila tai pop-up-kauppa)<sup>55</sup>. Kehitysprosessia ei välttämättä tarvitse – tai ole edes kannattavaa – viedä läpi vain yhden tyyppisiin kokeiluasetelmiin ja prototyyppihin turvautuen, vaan kokeilusyklien toistamisen myötä käytetyt kokeilukeinot voivat alkaa menettää tehoaan. Tällöin kokeilun tapojen muuttaminen voi parantaa prosessin etenemistä ja resurssien käyttöä.<sup>56</sup>

Koska alkuvaiheen kokeilujen tulisi olla helppoja ja nopeita toteuttaa olemassa olevien resurssien ja henkilökunnan varassa<sup>57</sup>, ei kokeiluasetelman ja prototyypin valmistelu saa vaatia suuria ajallisia ja taloudellisia panostuksia. Ensimmäisissä kokeiluissa käytettävien prototyyppien on oltava myös helposti muunneltavissa, jotta niitä voidaan parannella kokeilujen kautta saavutetun oppimisen mukaisesti<sup>58</sup>. Lisäksi konkretisointien viimeistelmättömyys on etu siitakin syystä, että koehenkilöt eivät ehkä uskaltaisi antaa rehellisesti negatiivista palautetta kovin valmiiksi hiotuista ratkaisuksista<sup>59</sup>. Idean kehittäjätäkään eivät välttämättä ole yhtä kiintyneitä karkeaan prototyyppiin kuin huolellisesti viimeistellyyn versioon, jolloin ideaa on helpompi muokata palautteen osoittamaan suuntaan<sup>60</sup>. Toisaalta kokeiluasetelman tulisi pystyä viestimään kehitettävää ideaa testaaville käyttäjille, millaista lisäarvoa idea heille tarjoaa. Koehenkilön kokemuksen tulisi olla mahdollisimman lähellä aitoa käyttötilannetta, jotta kokeilussa voitaisiin saada hyödyllistä palautetta ratkaisun toi-

53 Hassi ym. 2015, s. 106–121

54 Hassi ym. 2015, s. 134

55 Hassi ym. 2015, s. 136–170

56 Thomke 1998, s. 748–749

57 Anderson – Simester 2011, s. 102

58 Kälviäinen 2016

59 Kälviäinen 2016 ja Hassi ym. 2015, s. 135

60 Hassi ym. 2015, s. 135

mivuudesta.<sup>61</sup> Kokeiluasetelmaa suunniteltaessa on siis löydettävä tasapainotila mahdollisimman niukan resurssien käytön ja mahdollisimman aidon asiakaskokemuksen välillä<sup>62</sup>.

Prototyypin ohella kokeilun välineestä voidaan puhua ns. pienimpänä toimivana tuotteena (*minimum viable product, MVP*). Riesin määritelmän mukaan pienin toimiva tuote on kehitettävän idean versio, joka mahdollistaa kokeilusyklin kiertämisen pienimmällä mahdollisella ajalla ja vaivalla. Tällaisesta tuotteesta puuttuu useita ominaisuuksia valmiiseen tuotteeseen nähden, mutta se on riittävä kokeilujen suorittamiseen.<sup>63</sup> Keskeneneräisellä mallilla kokeilemisesta hyödytään, koska se mahdollistaa investointien suuntaamisen vain kokeilun kannalta relevantteihin idean osiin ja toisaalta tuotteen ominaisuuksien vähäisyys voi helpottaa kokeilun tulosten analysointia<sup>64</sup>.

#### **2.3.4 Kokeilun suorittaminen, tulosten analysointi ja jatkotoimista päättäminen**

Kokeilusuunnitelman laatimisen ja kokeiluasetelman valmistelun jälkeen kehitettävää ideaa testataan käytännössä. Kokeilussa toimimattomien ratkaisujen toivotaan paljastuvan jo varhaisessa vaiheessa, jottei resursseja tuhlataisi huonoon ideaan. Kokeilu, jossa idea osoittautuu epäonnistuneeksi, ei sinänsä ole epäonnistunut kokeilu – onhan se tuottanut kehitysprojektin kannalta hyödyllistä tietoa.<sup>65</sup>

Kun kokeilu on suoritettu, on aika analysoida kokeilussa kerätty tieto ja sen pohjalta päättää jatkotoimista. Tieto on syytä tallentaa ja pitää avoimesti organisaation jäsenten saatavilla, jotta samaa asiaa ei turhaan testattaisi uudestaan.<sup>66</sup> Kun kokeilujen myötä opitaan, mikä toimii ja mikä ei, voidaan kokeilun tuloksista johtaa ne käytännön tavat (toteuttamisen ideat), joilla kehitettävän idean potentiaali saadaan hyödynnettyä. Jokaisen iteraatio-syklin jälkeen prosessissa palataan kokeiltavan idean hienosäätämiseen kokeilujen kautta saadun tiedon mukaisesti.<sup>67</sup> Lopulta ainoastaan kokeiluissa menestyneiden ideoiden tulisi päätyä edelleen kehitettäväksi ja lopulta pilotoinnin, beta-testauksen tai ensimmäisen version lanseerauksen kohteiksi. Vähemmän menestyksellisten ideoiden kohdalla tulisi harkita pivottia, idean väliaikaista hyllyttämistä tai sen lopullista hylkäämistä.<sup>68</sup>

61 Hassi – Tuulenmäki 2012, s. 10

62 Hassi ym. 2015, s. 100

63 Ries 2016, s. 97. Pienin toimiva tuote voidaan nähdä prototyypistä poikkeavana käsitteenä, mutta Hassin ym. (2015) omaksumaan tapaan tässä tutkielmassa viitataan kokeilussa käytettävään oppimisen välineeseen prototyypinä. MVP-termi on noussut suosioon erityisesti Riesin edustaman Lean Startup -suuntauksen imussa, mutta termiä käytetään melko vaihtelevissa yhteyksissä. MVP:n, prototyypin ja beta-version eroista ks. esim. "Build an MVP, Not a Beta." Forte. <http://fresco.vc/build-an-mvp-not-a-beta>. 8.12.2017.

64 Thomke 1998, s. 745

65 Thomke 2001, s. 71–72 ja Anderson – Simester 2011, s. 105

66 Davenport 2009, s. 71

67 Hassi – Tuulenmäki 2012, s. 10

68 Davenport 2009, s. 71 ja Hassi ym. 2015, s. 69

### 3. IMMATERIAALIOIKEUDET OSANA YRITYKSEN STRATEGIAA

#### 3.1 IPR-strategian käsite

Sodankäynnistä liike-elämän tarkoituksiin omaksuttu strategian käsite on alkujaan viitanut sotajoukkojen asemoinnin suunnitteluun ennen vihollisen kohtaamista<sup>69</sup>. Käsitteen sisältö on käyttöyhteydestä toiseen mukautuessaan säilynyt siinä mielessä samana, että niin armeijan kuin yrityksenkin johdolle strategia merkitsee monimutkaista psykologisten toimintojen verkostoa, joka antaa suuntaa erilaisten toimien valinnassa tavoiteltaessa tiettyä päämäärää. Strategia määrittelee, miten tavoitteet saavutetaan, eikä strategiaa ole olemassa ilman ennalta hahmoteltuja tavoitteita. Vailla päämäärää suoritettut toimet ovat puhtaasti taktisia, eivät strategisia.<sup>70</sup>

Immateriaalioikeudet (*intellectual property rights, IPR*) ovat aineettomaan varallisuuteen kohdistuvia yksinoikeuksia, joihin vedoten niiden haltija voi kieltää muita hyödyntämästä kyseisiä oikeuksia. Perinteisesti immateriaalioikeudet on jaettu teollisoikeuksiin ja tekijänoikeuksiin.<sup>71</sup> Varsinaisten laissa säänneltyjen immateriaalioikeudellisten suojaustoimen lisäksi yritykset käyttävät aineettoman omaisuutensa suojaamiseen myös laissa sääntelemättömiä keinoja, kuten tiedon salaamista ja nopeita markkinaoperaatioita<sup>72</sup>. Jotta omaisuutta voitaisiin suojata tehokkaasti ja yksinoikeuksia käyttää kannattavalla tavalla, on immateriaalioikeudet ja muut suojaustavat otettava osaksi yrityksen toiminnan strategista suunnittelua<sup>73</sup>.

Organisaation kaikkia immateriaalioikeudellisia toimia ohjeistava immateriaalioikeusstrategia eli IPR-strategia sisältyy yrityksen laajempaan liiketoimintastrategiaan<sup>74</sup>. IPR-strategia muotoiltaan liiketoimintasuunnitelman kanssa yhdenmukaiseksi ja liiketoimintasuunnitelman muuttuessa myös IPR-strategia tulisi tarkastaa<sup>75</sup>. IPR-strategia määrittelee liiketoiminnan kannalta tärkeät elementit sekä keinot, joilla niitä voidaan suojata niin yrityksen sisäisessä kuin kolmansienkin kohdistuvassa toiminnassa<sup>76</sup>. Toimivan strategian muotoilemiseen vaikuttavat mm. yrityksen toimiala, koko, missio, kilpailu ja toimintaympäristö, joten erilaisten yritysten strategiatkin ovat erilaisia<sup>77</sup>.

69 Nickols 2016, s. 1–2

70 Nickols 2016, s. 7, 11

71 Mansala 2007, s. 588

72 Luoma – Huuskonen 2008, s. 16–17, 21

73 Mansala 2007, s. 588–589

74 Päällysaho – Kuusisto 2006, s. 13

75 Innanen – Jäske 2014, s. 242

76 von Willebrand 2012 ja Innanen – Jäske 2014, s. 246–247, 251

77 Gollin 2008, s. 163, Innanen – Jäske 2014, s. 240 ja Lahti 2006, s. 203

Karkeasti voidaan todeta, että IPR-strategian laatimisella halutaan taata yritykselle sen toiminnan kannalta tärkeät immateriaalioikeudet sekä turvata yritystä mahdollisilta oikeudenloukkauksilta<sup>78</sup>. Edellä kuvatun strategian käsitteen tavoin myös IPR-strategia on aina kytköksissä organisaation tavoitteisiin. Tavoitteet, joihin immateriaalioikeuksien käytöllä pyritään, ovat eri organisaatioiden kohdalla erilaisia, mutta voittoa tavoittelevat tahot pyrkivät toiminnassaan tyypillisesti varallisuutensa kasvattamiseen, sijoittajien houkuttelemiseen ja investointiensa suojaamiseen<sup>79</sup>. Immateriaalioikeuksien hankkiminen ja ylläpitäminen ei siis koskaan ole yritykselle itsetarkoitus, vaan immateriaalioikeuksia hyödyntämällä pyritään johonkin kaupalliseen päämäärään<sup>80</sup>.

### 3.2 IPR-strategian elementtejä

Koska suojaaminen kuluttaa yrityksen resursseja, on päätettävä, mitä kaikkea suojataan ja millä tavalla<sup>81</sup>. Laissa säänneltyjen immateriaalioikeuksien (muodolliset suojausmenetelmät) ohella aineetonta omaisuutta voidaan suojata sopimusten avulla (puolimoduolliset suojausmenetelmät) tai joustavilla, laissa sääntelemättömillä keinoilla (ei-muodolliset suojausmenetelmät<sup>82</sup>). Varsinaisia immateriaalioikeuksia hyödyntävät tyypillisesti suuret teollisuusyritykset, kun taas pienet ja keskisuuret yritykset suosivat muunlaisia suojauskeinoja.<sup>83</sup> Kulloisessakin tapauksessa sopivimman suojausmenetelmän valitseminen vaatii menetelmien ominaispiirteiden tuntemusta sekä ymmärrystä niiden soveltumisesta yrityksen tavoitteiden toteuttamiseen<sup>84</sup>.

Yrityksellä on oltava tietoa aineettomasta omaisuudestaan, jotta se voisi strategisesti arvioida erilaisten immateriaalioikeuksien käyttötapoja. Tunteakseen innovaatioprosessit, joissa aineetonta omaisuutta syntyy, on yrityksen kerättävä tietoa jo prosessien kuluessa. Tällaisen tiedon hankinta jälkikäteen on usein myöhäistä. Dokumentaation tehtävänä on selvittää luovan työn taustoja ja toimia todistusaineistona siitä, kuka on luovan työn alku-

78 Ashorn 2003, s. 85

79 Gollin 2008, s. 164

80 Sampson 2009, s. 411–412

81 Mansalan 2007, s. 600, mukaan yritykselle merkittävät kohteet kannattaa suojata vahvoin keinoin, kun taas merkitykseltään vähäisemmät tai lyhytkestoiset kohteet voidaan suojata heikommin. Em. tekijöiden ohella Innanen ja Jäske 2014, s. 242–243, mainitsevat myös kohteen kopioinnin helppouden suojaamisen puolesta puhuvana seikkana. Gollin 2008, s. 240, arvioi suojaamiseen investoimisen kannattavuutta sen kautta, miten hyvin suojattava kohde vastaa organisaation tavoitteisiin ja onko kohteella arvoa myös ulkopuolisille.

82 Pällysahon ja Kuusiston 2006, s. 4, mukaan ei-muodollisiin suojamenetelmiin kuuluvat mm. salassapito, idean julkaiseminen, nopea innovaatiotahti, tehokas tiedon jakaminen, tekniset suojausmenetelmät, monimutkainen tuotesuunnittelu, henkilöstön sitouttaminen ja asiakassuhteiden hoitaminen.

83 Pällysaho – Kuusisto 2006, s. 3–4, 15

84 Innanen – Jäske 2014, s. 193

peräinen tekijä.<sup>85</sup> IPR-portfolion sisällön hahmottamiseksi yrityksen on syytä pitää kirjallista rekisteriä kaikista toimintaansa liittyvistä immateriaalioikeuksista – sekä itse omistamistaan että käyttämistään kolmansien oikeuksista.<sup>86</sup> IPR-portfoliota tulee säännöllisesti tarkastaa, jolloin immateriaalioikeuksista aiheutuvia kuluja voidaan pyrkiä hallitsemaan luopumalla hyödyttömistä oikeuksista tai suojauksen tasoa heikentämällä<sup>87</sup>.

Jos yritys ei aio alusta asti itse tuottaa kaikkea aineetonta varallisuuttaan ja pitää kaikkia immateriaalioikeuksia jakamattomana itsellään, tulevat immateriaalioikeuksien hankkimista ja luovuttamista koskevat käytännöt määriteltäviksi yrityksen IPR-strategiassa. Immateriaalioikeuksien siirtäminen tapahtuu joko oikeuksien luovuttamisella (kaikki oikeudet siirtyvät) tai lisensioinnilla (kaikki oikeudet eivät siirry)<sup>88</sup>. Yhä useammin aineettoman varallisuuden tuottama taloudellinen hyöty kertyykin oikeuksien lisensioinnista, jolla tarkoitetaan käyttöoikeuden myöntämistä toisen omistamaan immateriaalioikeuteen<sup>89</sup>.

IPR-strategia sisältää yrityksen ulkoisia sopimuskäytäntöjä koskevan ohjeistuksen ohessa myös yrityksen sisäisen sopimisen säännöt. Jotta yritys saisi haltuunsa kaikki sen toiminnan kannalta keskeiset immateriaalioikeudet, on tärkeää ennalta suunnitella, miten työntekijöiden luovan toiminnan tulokset saadaan siirrettyä yrityksen nimiin. Tavallisesti työntekijä sitoutuu jo työsopimuksessa luovuttamaan työsuhteessa hänelle syntyvät immateriaalioikeudet työnantajalleen.<sup>90</sup> Sopimusten ohella yritys koordinoi sisäisiä immateriaalioikeusasioitaan organisaatiokulttuurinsa sekä moninaisten politiikkojen ja käytäntöjen kautta<sup>91</sup>. Esimerkiksi työsuhteidennovaatioasioita koskien strategian olisi hyvä sisältää paitsi oikeuksien siirtymistä koskevat käytännöt, mutta myös ohjeet aloitetoiminnasta, työsuhteidennovaatioiden käsittelystä sekä osapuolten oikeuksista ja velvollisuuksista<sup>92</sup>.

Myös oikeuksien valvonta ja puolustaminen on asia, johon tyypillisesti otetaan kantaa IPR-strategiassa. Yksinoikeuden haltijan on itse hoidettava oikeuksiensa valvominen, sillä mikään julkinen taho ei sitä viran puolesta suorita<sup>93</sup>. Yrityksen aktiivisuus oikeuksiensa valvonnan ja puolustamisen suhteen riippuu siitä, miten keskeinen osa sen strategiaa immateriaalioikeudet ovat ja millaista taloudellista arvoa niillä on. Jos immateriaalioikeudet

---

85 Innanen – Jäske 2014, s. 188–189 ja Sampson 2009, s. 415

86 Innanen – Jäske 2014, s. 57–59

87 Gollin 2008, s. 248–249 ja Innanen – Jäske 2014, s. 193

88 Gollin 2008, s. 268

89 Mansala 2001, s. 2, 4

90 Gollin 2008, s. 268 ja Innanen – Jäske 2014, s. 249–250

91 Gollin 2008, s. 147

92 Päällysaho – Kuusisto 2006, s. 13

93 Innanen – Jäske 2014, s. 59

ovat yritykselle tärkeitä, sen on suoritettava tarkkaa valvontaa ja puututtava herkemmin havaitsemiinsa loukkauksiin. Jos aineeton varallisuus taas ei ole yritykselle niin arvokasta, että sitä kannattaisi suojella, on yrityksen mahdollisesti järkevintä keskittyä vain välttämään toisten oikeuksien loukkaamisesta johtuvia velvoitteita. Immateriaalioikeuksien puolustamisen tulisi olla suhteessa oikeuksien taloudelliseen potentiaaliin, ja yrityksen kannattaa jo ennalta harkita kalliille tuomioistuinprosessille vaihtoehtoisia tapoja puuttua oikeudenloukkauksiin. Tavallisimmin loukkausepäily selvitetään neuvotteluin, hallinnollisissa prosessissa, oikeudenkäynnissä tai välimiesmenettelyssä, joskin oikeudenkäyntiin turvaudutaan vasta viimesijaisena keinona.<sup>94</sup>

### 3.3 Tekijänoikeuden erityispiirteitä IPR-strategian kannalta

Tekijänoikeuden ydinalojen yhteenlaskettu arvonnäisyys vuonna 2012 oli noin 7,3 miljardia euroa eli noin 3,8 prosenttia Suomen bruttokansantuotteesta<sup>95</sup>, ja omana teollisuudenalana voidaan puhua kokonaisesta tekijänoikeusteollisuudesta. Tekijänoikeus ei siis ole liiketoiminnassa teollisoikeuksiin nähden mitenkään toissijainen immateriaalioikeus, vaan myös tekijänoikeudet voivat tuottaa yritykselle tuloja ja kilpailuetua. Tekijänoikeudella on kuitenkin teollisoikeuksiin nähden tiettyjä erikoispiirteitä, jotka vaikuttavat sen niveltymiseen osaksi yrityksen IPR-strategiaa.

Tekijänoikeuden alaan kuuluvien kohteiden tapauksessa ei ole tarpeen puntaroida, tulisiko kohde suojata ja miten pian suojaa tulisi hakea. Tekijänoikeuslain (TekijäL, 8.7.1961/404) 1 § ei nimittäin aseta tekijänoikeuden syntymiselle muita edellytyksiä kuin teoksen luomisen, ja tekijänoikeutta koskevat konventiot kieltävät muotovaatimusten asettamisen tekijänoikeussuojan saamisen edellytykseksi<sup>96</sup>. Tekijänoikeus siis syntyy tekijälle lain nojalla automaattisesti sillä hetkellä, kun hän luo teoksen, ja yksinoikeus on voimassa verrattain pitkään: peräti 70 vuotta tekijän kuolinvuodesta (TekijäL 43 §). Koska tekijänoikeussuojan saaminen ei teollisoikeuksien tavoin vaadi kuluja aiheuttavaa rekisteröimisprosessia eikä oikeuden voimassa pitäminen edellytä vuosimaksujen suorittamista, ei pelkkä tekijänoikeuksien pitäminen osana yrityksen IPR-portfoliota sinänsä kuluta yrityksen resursseja. Yritykselle hyödyttömästä tekijänoikeudesta ei siten ole taloudellisista syistä tarpeen hankkiutua eroon, kuten teollisoikeuksien kohdalla suositellaan. Toisaalta tekijänoikeuk-

<sup>94</sup> Mansala 2007, s. 600 ja Sampson 2009, s. 416

<sup>95</sup> Grönlund ym. 2014, s. 4

<sup>96</sup> Muotovaatimusten kieltö ilmaistaan esimerkiksi Bernin konvention 5 artiklassa.



sienkin valvomisesta ja puolustamisesta aiheutuu kuluja.

Myöskään suojauksen alueellinen ulottuvuus ei tekijänoikeuden kohdalla ole samanlainen strategista harkintaa vaativa kysymys kuin teollisoikeuksien tapauksessa. Teollisoikeudet luovat haltijalleen pääsääntöisesti yksinoikeuden vain sellaisissa valtioissa, joissa oikeus on saatettu voimaan rekisteröimällä, kun taas teoksen saama tekijänoikeussuoja ei ole samalla lailla maantieteellisesti rajattu. Kansainvälisillä tekijänoikeussopimuksilla useat valtiot ovat sitoutuneet takaamaan kansallisessa lainsäädännössään tietyn tekijänoikeudellisen minimitasen (vähimmäissuojan periaate) ja antamaan toisten sopimusvaltioiden kansalaisten teoksille samaa tekijänoikeussuojaa kuin omille kansalaisilleen (kansallisen kohtelun periaate)<sup>97</sup>. Koska lukuisat maat ovat liittyneet ainakin johonkin tekijänoikeuskonventioon, on tekijänoikeussuoja käytännössä voimassa varsin laajalla alueella. Esimerkiksi merkittävimmän tekijänoikeuskonvention eli Bernin yleissopimuksen<sup>98</sup> osapuolina on tällä hetkellä 175 valtiota<sup>99</sup>.

Tekijänoikeus ei välttämättä ole ainoa keino suojata aineetonta omaisuutta, joka on TekijäL 1 §:n mukainen teos. Suurin hyöty yleensä saadaankin käyttämällä erilaisia suojaamisen menetelmiä yhdessä<sup>100</sup>. Immateriaalioikeudellisten suojamuotojen välisten rajojen on jossain määrin katsottu hämärtyneen, mikä mahdollistaa eri immateriaalioikeuksien kumuloitumisen samaan kohteeseen. Esimerkiksi tavaramerkki- ja tekijänoikeussuojan päällekkäisyys tulee usein kyseeseen molempien suojamuotojen viimeaikaisen laajentumisen johdosta.<sup>101</sup> TekijäL 10.1 §:ssä puolestaan nimenomaisesti säädetään mahdollisuudesta suojata tuotetta sekä teoksena että rekisteröitynä mallina. Sopimusperustaisia tai ei-muodollisia suojaamisen menetelmiä voidaan luonnollisesti niin ikään soveltaa teoksiin.

97 Harenko ym. 2016, s. 9

98 Bernin yleissopimus kirjallisten ja taiteellisten teosten suojaamisesta, SopS 79/1986

99 "WIPO-Administered Treaties." World Intellectual Property Organization. [http://www.wipo.int/treaties/en/ShowResults.jsp?lang=en&treaty\\_id=.15](http://www.wipo.int/treaties/en/ShowResults.jsp?lang=en&treaty_id=.15). 22.12.2017.

100 Innanen – Jäske 2014, s. 247 ja Mansala 2001, s. 8

101 Pitkänen 2016, s. 42, 59



## 4. KOKEILUASETELMA TEKIJÄNOIKEUDELLISESTA NÄKÖKULMASTA

### 4.1 Tekijänoikeuden oikeuttamisperusteista

Tekijän oikeusasema muodostuu kahtalaisista oikeuksista: taloudellisista (TekijäL 2 §) ja moraalisisista (TekijäL 3 §). Taloudellisten oikeuksien perusteluna on käytetty John Locken työteoriaan<sup>102</sup> pohjautuvaa ajatusta tekijän työpanoksen luomasta omistusoikeudesta ja luomistyön palkitsemisesta, kun taas moraalisten oikeuksien tarkoituksena on suojata tekijän persoonaa.<sup>103</sup> Tekijäyksilön intresseihin kiinnittyvien oikeuttamisperusteiden ohessa tekijänoikeutta perustellaan yleensä myös yhteisöllisemmillä näkökohdilla. Ns. kannustavuusargumentin mukaan tekijänoikeuden tekijälle tuottama yksinoikeusasema kannustaa yksilöitä luovaan toimintaan, mikä taas johtaa tieteiden ja taiteiden kehitykseen koko yhteiskunnan tasolla. Tekijänoikeuden tuottamaa yleistä etua painottavassa utilitaristisessa teoriassa tekijän ja yhteisön etu eivät näyttäydy toistensa vastapoolleina, vaan yksilön hyvä tuottaa myös yhteistä hyvää.<sup>104</sup> Oikeuskirjallisuuden lisäksi tekijänoikeuden kannustinvaikutukseen ja utilitaristiseen ajattelumalliin viitataan niin kansallisen tekijänoikeuslain esitöissä kuin kansainvälisissä tekijänoikeussopimuksissakin<sup>105</sup>.

Kuitenkin tekijänoikeudellisessa keskustelussa tekijänoikeuden haltijan ja yleisön etujen on nähty olevan epätasapainossa tai jopa ristiriidassa. Kriitikoiden mukaan tekijänoikeuden liiallinen rinnastaminen perinteiseen omaisuuteen on tehnyt tekijänoikeudesta lähtökohtaisesti vahvan yksinoikeuden, joka toimii pikemminkin kontrollin välineenä kuin innovaatiopoliittisena porkkanana. Fyysisen omaisuuden kaltaiseksi ymmärrettynä tekijänoikeuden on pelätty irtautuvan luovaa työtä edistävästä tavoitteistaan ja alkavan palvella lähinnä suppeiden ryhmien taloudellisia intressejä.<sup>106</sup> Kannustavuusargumentin paikkansapitävyyttä ylipäänsä on epäilty, ja vähintäänkin voidaan perustellusti asettaa kyseenalaiseksi, onko kaikkein toimivin kannustin juuri yksinoikeus teoksen taloudelliseen hyödyntämiseen<sup>107</sup>.

102 Locken mukaan tekijälle syntyy tekemänsä työn kautta omistusoikeus työnsä tuloksiin, mutta myös muiden käyttöön on jätettävä riittävästi ja yhtä hyvää materiaalia luovan työn raaka-aineeksi. Locke 1982, s. 17–20. Työteoriaa on käytetty perustelemaan paitsi omistusoikeutta, myös tekijänoikeuden rajoittamistarvetta ja materiaalien yhteiskäyttöön jättämistä. Ks. Riekkinen 2007, s. 192 ja Leppämäki 2007, s. 85.

103 Haarmann 1992, s. 88 ja Leppämäki 2007, s. 81

104 Leppämäki 2007, s. 81, Mylly 2004, s. 229–230 ja Riekkinen 2007, s. 201–202

105 Esimerkiksi TRIPS-sopimuksen johdanto-osassa immateriaalioikeuksien tavoitteeksi todetaan yhteiskunnan kehittäminen, ja WIPO:n tekijänoikeussopimuksen johdannossa painotetaan tekijänoikeuden merkitystä luovuuden kannustimena. Esitöistä ks. esim. HE 170/1997 vp., s. 4.

106 Ks. Lavapuro 2007, s. 154, 158, 165, 167 ja Riekkinen 2007, s. 174

107 Ks. Lessig 2001, s. 71 ja Riekkinen 2007, s. 196, 217–218

Perusoikeusnäkökulmasta tekijänoikeus kuuluu perustuslain (PL, 11.6.1999/731) 15 §:ssä säännellyn omaisuuden suojan piiriin, ja tekijän yksinoikeutta voidaan perustella myös PL 18 §:n mukaisella elinkeinovapaudella<sup>108</sup>. Toisaalta tietyt perusoikeudet, kuten sananvapaus (PL 12 §) ja sivistykselliset perusoikeudet (PL 16 §), näyttävät edustavan tekijänoikeuden kanssa osin samansuuntaisia ja osin vastakkaisia päämääriä. Nykyistä tekijänoikeusdoktriinia on kritisoitu liiallisesta tekijäkeskeisyydestä, joka jättää varjoonsa yleisön oikeudet. Vaikka tekijän yksinoikeutta henkisen työnsä tulokseen suojataan PL 15 §:n nojalla, ei omaisuuden suoja saa muihin perusoikeuksiin nähden minkäänlaista etusijaa, vaan moniarvoisessa yhteiskunnassa on muitakin samanarvoisia perustavanlaatuisia oikeuksia.<sup>109</sup> Lisäksi tekijänoikeuden perustelu yksilön oikeuksien suojaamisella on nähty siinä mielessä harhaanjohtavana, että nykyään varsin usein tekijänoikeuden haltija on tekijämyytin mukaisen luovan yksilön sijasta modernin tekijänoikeusteollisuuden suuryritys<sup>110</sup>.

Tekijänoikeuden on digitalisaation ja immateriaalioikeuksien kasvavan taloudellisen merkityksen myötä väitetty laajentuneen ja vahvistuneen<sup>111</sup>. Vaikka laajaa tekijänoikeussuojaa on perusteltu sen luovaan toimintaan kannustavalla vaikutuksella, voidaan tekijänoikeuden alan kasvamisen arvella vaikuttavan myös päinvastaisella tavalla: kun tekijänoikeuden suojaama alue leviää, kaikkien vapaasti käytettävissä oleva materiaali vähenee. Vapaan materiaalin vähäisyys vaikuttaa luovuuteen negatiivisesti, koska tekijä ei luo töitään tyhjästä, vaan uudet innovaatiot syntyvät usein aiempien henkisen työn tuotteiden pohjalta. Tekijänoikeuden laajentumisen on ajateltu muodostavan luovuudelle esteitä myös tekijänoikeudellisten loukkausten riskien ja lisenssien hankkimiseen kulumien investointien kasvun muodossa.<sup>112</sup> Vaikka tekijänoikeussuojan oikeuttamisessa käytetään argumenttia tekijänoikeuden rakenteellisesta demokratiavaikutuksesta, saattaa laajan tekijänoikeussuojan vaikutus olla päinvastainen ja epätasa-arvoistava<sup>113</sup>.

Toisaalta oikeustieteellisessä kirjoittelussa tekijänoikeuteen kohdistettua kritiikkiä on pidetty osin liioiteltuna<sup>114</sup>. Tekijänoikeus ei ole absoluuttinen yksinoikeus, vaan se on asialli-

108 Haarmann 2014, s. 48 ja Koillinen – Lavapuro 2002, s. 343

109 Ks. Koillinen – Lavapuro 2002, s. 346, Lavapuro 2007, s. 161 ja Leppämäki 2007, s. 82–83. Vrt. Oesch 2005, s. 361–362, jonka mukaan tekijänoikeuslainsäädännön laatiminen ensisijassa tekijän edun näkökulmasta on luonteva ratkaisu, koska tekijä on yleisöön nähden heikompi osapuoli ja tekijän etujen suojaaminen koituu myös yhteiskunnan hyväksi. Lisäksi Oeschin mukaan tekijän ja yleisön perusoikeuksien tasapainon etsintä on vakiintunut osa lainvalmistelua.

110 Ks. Koillinen – Lavapuro 2002, s. 347–348, Lavapuro 2007, s. 162 ja Mylly 2004, s. 238

111 Pihlajarinne 2014, s. 867, Koillinen – Lavapuro 2002, s. 340 ja Riekkinen 2007, s. 181. Vrt. Oesch 2005, s. 361.

112 Lessig 2001, s. 12 ja Mylly 2004, s. 234

113 Mylly 2004, s. 230–231, 251 ja Riekkinen 2007, s. 215

114 Ks. esim. Oesch 2005, s. 357 ja Oesch 2017, s. 2

sesti ja ajallisesti rajoitettu<sup>115</sup> – joskin tekijänoikeussuojan kestoa voidaan ainakin tiettyjen teoslajien, esimerkiksi tietokoneohjelmien, kohdalla perustellusti pitää liian pitkänä. Tekijän yksinoikeuden laajuutta supistavat myös TekijäL 2 luvun rajoitussäännökset, joiden tarkoituksena on turvata erilaisia yhteiskunnallisia tarkoituksia ja perusoikeuksia. Kuitenkin myös 2 luvun rajoitusten kohdalla tekijänoikeuden haltijan ja yleisön oikeuksien tasapaino-ongelma ilmenee rajoitusten tulkintatapaa koskevassa keskustelussa<sup>116</sup>. Tekijän ja yleisön edut eivät ole sovittamattomassa ristiriidassa keskenään, vaan tekijänoikeuden taustalla voidaan nähdä esimerkiksi sananvapautta edistäviä ja sivistyksellisiä päämääriä. Varsinainen ongelmakohta ei näyttäisi olevan tekijänoikeusinstituutio itsessään, vaan tekijänoikeuden suoja-alan ideaalin laajuuden määrittäminen.<sup>117</sup>

## 4.2 Tekijänoikeuden kohde

### 4.2.1 Teostas

TekijäL 1 § osoittaa tekijänoikeuden kohteeksi kirjallisen tai taiteellisen teoksen, jonka tekijä on luonut. 1 § sisältää myös luettelon tyypillisistä teoslajeista, mutta luettelo ei ole tyhjentävä, vaan teos voi ilmetä muullakin kuin lainkohdassa mainitulla tavalla<sup>118</sup>. Komiteamietinnössä 1953:5 todetaankin, ettei teoksen täsmällistä määrittelyä laissa katsottu tarpeelliseksi tai edes mahdolliseksi, vaan luomuksen kuuluminen tekijänoikeussuojan alaisuuteen ratkaistaan aina konkreettisen tapauksen erityispiirteiden mukaan<sup>119</sup>. Tekijänoikeuslaki ei siis anna kovin tarkkarajaista sisältöä teoksen käsitteelle, vaan sen yksityiskohtaisempi muotoilu on jäänyt oikeustieteen ja -käytännön tehtäväksi.

Oikeuskirjallisuudessa käytetään vakiintuneesti teoskynnyksen tai -tason käsitettä apuna sen määrittelemiseksi, mitkä tuotteet ovat teoksia ja mitkä eivät. Kun henkisen luomistyön tuote ylittää teoskynnyksen eli yltää teostasoon, sen katsotaan kuuluvan tekijänoikeuslain soveltamisalaan ja siten saavan tekijänoikeussuojaa. Vaikka teos sinänsä voi ilmetä missä tahansa muodossa, on teoskynnys joidenkin teoslajien kohdalla korkeampi kuin

<sup>115</sup> Oesch 2017, s. 6

<sup>116</sup> 2 luvun rajoitusten ilmentämiin perusoikeuksiin vedoten voidaan argumentoida, ettei rajoituksia tulisi tulkita suppeasti, vaan tulkinnan tulisi lähteä eri intressitahojen perusoikeuksien punninnasta. Ks. Koillinen – Lavapuro 2002, s. 344, Mylly 2004, s. 243 ja Pihlajarinne 2014, s. 868. Vrt. Oesch 2005, s. 360, jonka mukaan tekijänoikeuden rajoitusten suppea tulkintatapa ei ole perusoikeuksien rajoittamisperiaatteiden kanssa ristiriidassa eikä aiheuta ongelmia käytännön tulkintatilanteissa.

<sup>117</sup> Ks. Pihlajarinne 2014, s. 867 ja Koillinen – Lavapuro 2002, s. 343

<sup>118</sup> KM 1953:5, s. 44. TekijäL 1 §:n teoslajien luettelon avoimuudesta kertoo jo luettelon loppuun otettu ilmaus ”–tai ilmetköönpä se (teos) muulla tavalla”. Tällä viitataan myös siihen, että tekijänoikeus teokseen säilyy silloinkin, kun teos toisinnetaan toisessa teoslajissa. Ks. Kivimäki 1966, s. 21.

<sup>119</sup> KM 1953:5, s. 44

toisten. Kirjallisten teosten, sävellysteosten ja kuvataiteen on tavallisesti katsottu yltävän teostasoon varsin helposti, kun taas taideteollisuuden tuotteiden kohdalla teoskynnyksen ylittäminen on vaikeampaa. Kuitenkin tuomioistuimen on lopulta aina arvioitava teostaso yksittäistapauksessa eli ratkaistavana olevassa riita- tai rikosasiassa.<sup>120</sup>

Teoskynnyksen ylittääkseen tuotteen on oltava itsenäinen ja omaperäinen. Vaikka itsenäisyyden ja omaperäisyyden vaatimusta ei nimenomaisesti laissa mainita, kriteerin katsotaan sisältyvän TekijäL 1 §:n ilmauksiin ”luonut” ja ”teos”<sup>121</sup>. Itsenäisyyden vaatimus tarkoittaa sitä, ettei teosta ole mahdollista luoda vain aiemmin luotuja teoksia jäljittelemällä<sup>122</sup>. Teoksen tulee siis olla tekijälle subjektiivisesti uusi, mutta esimerkiksi patentoitavalta keksinnöltä edellytettävää objektiivista uutuutta ei teostasoon yltäminen vaadi<sup>123</sup>. Omaperäisyys puolestaan määritellään ns. kaksoisluomiskriteerin avulla: kyseessä on teos, jos kukaan toinen vastaavaan luomistyöhön itsenäisesti ryhtyessään ei olisi päätenyt samaan lopputulokseen<sup>124</sup>. Myös teoksen osa voi saada tekijänoikeussuojaa, jos se kokonaisuudesta irrotettunakin ylittää teoskynnyksen<sup>125</sup>.

Teoksen tekijänoikeudellinen arviointi ei ole teoksen taiteellisen laadun arviointia, vaan verrattain heikkotasoinenkin henkisen työn tuote voi yhtä lailla yltää teostasoon kuin esteettisesti korkeatasoinen tuote. Tekijänoikeussuojaa ei Suomessa saa myöskään eräissä maissa tunnetulla ns. skill and labour -perusteella<sup>126</sup>, eli tuotteen luomisen vaatimalla taidon ja resurssien määrällä ei ole teosarvioinnin kannalta merkitystä.<sup>127</sup>

#### **4.2.2 Muodon ja idean dikotomia**

Tekijänoikeuden kohdetta on eri aikoina luonnehdittu eri tavoin. Varhaisemman käsityksen mukaan suojan kohteena olivat tekijän persoonalliset ideat (subjektiivinen ideaoppi), mutta nykyaikaisemman katsantokannan mukaan tekijänoikeuden antama yksinoikeus kohdistuu teokseen sen abstraktissa muodossa (objektiivinen ideaoppi).<sup>128</sup> Teoksen käsite ei siis samaistu teoskappaleen käsitteeseen, vaan sama teos voi jopa ilmetä eri teoslajeissa<sup>129</sup>.

<sup>120</sup> Haarmann – Mansala 2012, s. 90 ja Harenko ym. 2016, s. 18

<sup>121</sup> KM 1953:5, s. 44

<sup>122</sup> Haarmann – Mansala 2012, s. 89–90 ja Harenko ym. 2016, s. 17

<sup>123</sup> Haarmann 1992, s. 55 ja Haarmann 2014, s. 57. Komiteamietinnössä 1953:5, s. 44, teoksen edellytetään olevan ”uusi”, mutta tämä on tulkittu objektiivisen uutuuden vaatimuksen sijasta lähinnä itsenäisyyden vaatimuksen synonyymiksi. Ks. Haarmann 1992, s. 53.

<sup>124</sup> Haarmann 1992, s. 54–55

<sup>125</sup> Harenko ym. 2016, s. 19

<sup>126</sup> Poikkeuksen muodostaa investointien suojaksi tulkittu TekijäL 49 §:n ilmaisema tietokantojen sui generis -suoja. Ks. esim. Harenko ym. 2016, s. 497–506.

<sup>127</sup> Haarmann – Mansala 2012, s. 89–90, Harenko ym. 2016, s. 17 ja Pihlajarinne 2014, s. 879

<sup>128</sup> Haarmann 1992, s. 44–45

<sup>129</sup> Kivimäki 1966, s. 21

Kuitenkin konkreettisessa tapauksessa teos tunnustetaan sen aineellisten ilmentymien, kuten teoskappaleiden, kautta<sup>130</sup>. Komiteamietinnössä 1953:5 ilmaistu käsitys tekijänoikeuden kohteesta näyttäisi tukeutuvan lähinnä objektiiviseen ideaoppiin, sillä mietinnössä tekijänoikeuden todetaan koskevan ”ensisijassa henkistä eli aineetonta luomusta”<sup>131</sup>.

Tekijänoikeutta kutsutaan monesti muodon suojaksi, sillä suoja ei kata esimerkiksi teoksen ideaa, aihetta, teemaa, juonta tai teoksen sisältämää informaatiota<sup>132</sup>. Tällaiset henkisen työn tuotteeseen liittyvät abstraktisemmat ominaisuudet on rajattu tekijän yksinoikeuden ulkopuolelle, jotta myös muut voisivat hyödyntää niitä omassa luovassa toiminnassaan. Lainsäätäjä ei siis ole olettanut luomisen tapahtuvan jonkinlaisessa tyhjiössä, vaan toisten teoksista inspiroituminen on täysin sallittua. Tätä rajanvetoa sallitun teoksen sisältämän idean käyttämisen ja kielletyn teoksen muodon kopioimisen välillä ilmentää myös TekijäL 4.2 §, jonka mukaan toisen teosta vapaasti muuttaen uuden ja itsenäisen teoksen aikaansaanut henkilö saa teokseensa alkuperäisteoksen tekijästä riippumattoman tekijänoikeuden (ns. vapaa muunnelma).

Jos teos pyritään hahmottamaan suojatun muodon ja suojaamattoman idean kaksijaon näkökulmasta, olisi kyettävä määrittelemään, mitkä teoksen elementit lukeutuvat sen muodon ja mitkä sen idean piiriin. Tällaisen erottelun tekeminen voi kuitenkin olla varsin hankalaa. Kivimäen mukaan teoksen arvioinnissa suurin painoarvo tulisi antaa teoksen muodolle, mutta hän toteaa muodon ja idean erillisen tarkastelun olevan tiettyjen teoslajien, kuten sävellysteosten, kohdalla mahdotonta. Näin ollen yksittäistapauksessa huomiota tulisi kiinnittää myös teoksen ”sisältöön”.<sup>133</sup> Myllyn mielestä nykyinen tekijänoikeus ei kuitenkaan ole luonnehdittavissa muodon suojaksi, koska teoksen ”abstraktin muodon” suojaaminen tarkoittaa käytännössä ainakin teoksen hienostuneimpien ideoiden suojaamista<sup>134</sup>. Haarmann taas päätyy eräänlaiseen kompromissiin: hänen mukaansa teos on ”jotakin, joka sijoittuu sisällön ja muodon väliin”<sup>135</sup>.

Tekijänoikeuden kohteen tieteellisiä määrittely-yrityksiä on pidetty siinä mielessä kritiikille alttiina, ettei abstrakti teos ole empiirisesti havainnoitavissa. Tekijän ja tekijänoikeuden kohteen välisen suhteen tarkastelun sijasta tekijänoikeus voidaankin hahmottaa myös eri henkilötahojen välisinä suhteina. Tällöin tekijänoikeuden kohde näyttäytyy fiktiivisenä entiteettinä, jolla on lähinnä esitystekninen merkitys.<sup>136</sup>

130 Haarmann 1992, s. 45

131 KM 1953:5, s. 44

132 Haarmann – Mansala 2012, s. 91, Harenko ym. 2016, s. 17 ja Kivimäki 1966, s. 18

133 Kivimäki 1966, s. 18–19

134 Mylly 2004, s. 241

135 Haarmann 2014, s. 67

136 Pihlajarinne 2014, s. 860–861

## 4.3 Tietokoneohjelman prototyyppi teoksena

### 4.3.1 Ohjelmakoodi ja tekijänoikeus

Tietokone voidaan karkeasti jakaa kahteen peruselementtiin: fyysiseen laitteistoon (*hardware*) sekä ohjelmistoon (*software*), joista jälkimmäinen pitää sisällään tietokoneohjelmia (*program*) ja muunlaista ohjelmien oheismateriaalia.<sup>137</sup> Tekijänoikeuslain esitöissä tietokoneohjelma määritellään joukoksi käskyjä, jotka saavat tietokoneen suorittamaan halutun toiminnon<sup>138</sup>. Tietokoneohjelmien kuuluminen tekijänoikeussuojan alaisuuteen vahvistettiin lainmuutoksella 34/1991, mutta jo tätä ennenkin ohjelmia voitiin suojata tekijänoikeudella<sup>139</sup>. TekijäL 1.2 § mukaan tietokoneohjelmaa suojataan kirjallisena teoksena, ja sama kanta ilmaistaan myös ohjelmistodirektiivissä<sup>140</sup> sekä kansainvälisissä tekijänoikeussopimuksissa<sup>141</sup>. Ratkaisu perustuu siihen, että ohjelma luodaan kirjoittamalla se tiettyä koodistoa eli ohjelmointikieltä käyttäen<sup>142</sup>. TRIPS-sopimuksen<sup>143</sup> 10 artiklan 1 kohdan mukaan suoja ulottuu sekä ohjelman lähde- että konekieliseen muotoon. Ohjelman lähdekoodi (*source code*) on ihmisen ymmärtämä koodi, jonka ohjelmoija kirjoittaa jollain ohjelmointikielellä. Kun ohjelmoija on saanut lähdekoodin kirjoitettua, se muunnetaan konekieliseksi binäärikoodiksi (*object code*) eli tietokoneen ymmärtämäksi nollien ja ykkösten joukoksi.<sup>144</sup>

Ohjelmistodirektiivin 1 artiklan 2 kohdassa tietokoneohjelmien tekijänoikeudellinen suoja rajataan koskemaan ohjelman eri ilmaisumuotoja ja ohjelman perustana olevien ideoiden ja periaatteiden todetaan jäävän suojan ulkopuolelle. Ilmaisumuodoksi luetaan ohjelmakoodi sellaisessa muodossa, johon ohjelmoija on sen kirjoittanut, joten aiempaa ohjelmaa ulkoisesti muistuttavan uuden ohjelman luominen voi olla täysin tekijänoikeuslain mukaista – kunhan ohjelmien koodi vain on erilainen<sup>145</sup>. Näin ollen tekijänoikeus suojaa tietokoneohjelmaa lähinnä koodin suoralta kopioimiselta, mikä Takin mukaan merkitsee suojan alkamista ”siitä, mihin taloudellisen suojan tarve loppuu”<sup>146</sup>. Tietokone-

<sup>137</sup> Heinonen 1998, s. 12

<sup>138</sup> HE 161/1990 vp, s. 50

<sup>139</sup> Haarmann – Mansala 2012, s. 89

<sup>140</sup> Euroopan parlamentin ja neuvoston direktiivi 2009/24/EY tietokoneohjelmien oikeudellisesta suojasta

<sup>141</sup> Ks. Ohjelmistodirektiivin 1 art. 1 kohta, TRIPS-sopimuksen 10 art. 1 kohta ja WIPO:n tekijänoikeussopimuksen 4 art. Tietokoneohjelmien lukemista kirjallisiksi teoksiksi on kuitenkin pidetty ongelmallisena, sillä lähdekoodi on vain vaikein ihmisen ymmärrettävissä (konekielinen koodi ei lainkaan) ja ohjelmat käytännössä näyttäytyvät ihmisille muussa kuin kirjallisessa muodossa. Ks. Goldstein – Hugenholtz 2013, s. 212.

<sup>142</sup> Oksanen 2004

<sup>143</sup> Sopimus teollis- ja tekijänoikeuksien kauppaan liittyvistä näkökohdista, SopS 5/1995

<sup>144</sup> Abid 2005, s. 819–820 ja Takki 2003, s. 44

<sup>145</sup> Innanen – Jäske 2014, s. 143

<sup>146</sup> Takki 2003, s. 62. Tekijänoikeussuojan ulkopuolelle jäävät ohjelman rakenne ja systeemi-idea voivat olla ohjelmistokehityksen arvokkaimpia osia, joiden tunteminen mahdollistaa ulkopuolisille tahoille vastaavan sovelluksen kehittämisen ilman koodin kopiointiakin.

ohjelmana tekijänoikeussuojaa eivät voi saada ohjelman yhteydessä käytetty ohjelmointikieli tai tiedostomuoto<sup>147</sup>. Suojan saamisen kannalta ei myöskään ole merkityksellistä, millaiselle alustalle (kuten paperille, levykkeelle tai tietokoneen keskusmuistiin) koodi on tallennettu<sup>148</sup>. Koska ohjelma on pelkkien esteettisten arvojen toteuttamisen sijasta luotu toteuttamaan tietty tarkoitus, sulautuvat ohjelman idea ja ilmaisu usein toisiinsa. Ohjelman suojaamattoman idean ja suojattavan muodon ala jääkin yksittäistapauksessa ratkaistavaksi.<sup>149</sup>

Tekijänoikeusneuvoston lausunnossa 2008:13 musiikin soittamiseen käytettyjen tietokoneohjelmien katsottiin olevan itsenäisesti luotuja teoksia ohjelmien samanlaisesta toiminta-ajatuksista huolimatta. Ratkaisevaa arvioinnissa oli se, ettei ohjelmien lähdekoodien vertailu paljastanut selkeitä merkkejä koodin kopioimisesta. Neuvoston asiassa ottama kanta ilmentää tekijänoikeuden luonnetta ilmaisumuodon suojana; vaikka tietokoneohjelmiin sisältyi samankaltainen idea, ohjelmat nauttivat tekijänoikeussuojaa itsenäisinä teoksina, sillä ohjelmien lähdekoodit olivat toisistaan riippumattomasti toteutettuja. Teoksen muotoa ja ideaa kosketi myös lausunto TN 2011:15, jossa lausunnon hakijan kuvailema useissa laitteissa toimivan rahapelin toiminta-ajatus katsottiin tekijänoikeussuojan ulkopuolelle jääväksi ideaksi. Ohjelmakoodin esittämisen sijasta hakija oli vain kuvaillut neuvostolle pelin toiminnallisuutta, joka ei voinut saada tekijänoikeussuojaa.

Ohjelmistodirektiivin 1 artiklan 1 kohdan mukaan tietokoneohjelman käsite kattaa myös valmistelevaan suunnitteluun liittyvän aineiston. Direktiivin johdanto-osassa täsmennetään valmistelevan suunnittelutyön suojan koskevan luonteeltaan sellaisia toimia, joiden tuloksena voi myöhemmässä vaiheessa olla tietokoneohjelma. Valmistelevaan suunnitteluun liittyvä dokumentti ei kuitenkaan saa tekijänoikeussuojaa, jos se on niin yleisluontoinen, että se ilmaisee lähinnä ohjelman idean. Toisaalta vuokaavio tai muu yksityiskohtaisempi dokumentti voisi myös täysin samastua tietokoneohjelmaan, jos ohjelman lähdekoodi riittävän tarkasti ilmenisi dokumentista.<sup>150</sup>

Kuten kaikkien erilaisten teosten, myös tietokoneohjelman on esitöiden mukaan ylitettävä teoskynnys, jotta se voisi saada tekijänoikeuslain mukaista suojaa<sup>151</sup>. Myös tekijänoikeusneuvosto on lausunnossa 2005:7 todennut teostasovaatimuksen tulevan sovellettavaksi

---

147 Ks. EUT C-406/10 (SAS Institute)

148 HE 161/90 vp., s. 50

149 Takki 2003, s. 41 ja Välimäki 2009, s. 20–21

150 Koivumaa 1995, s. 62–63

151 HE 161/90 vp., s. 16



tietokoneohjelmien kohdalla. Myöhemmin tekijänoikeusneuvosto on kuitenkin esimerkiksi lausunnoissa 2006:5 ja 2006:12 todennut tietokoneohjelmien kohdalla teoskynnyksen olevan varsin alhainen. Suomessa tietokoneohjelmille asetettua teostasovaatimusta onkin kritisoitu sen vuoksi, että ohjelmistodirektiivin 1 artiklan 3 kohdan on katsottu edellyttävän teoskynnyksen madaltamista<sup>152</sup>.

Teokselta vaadittava itsenäisyys ja omaperäisyys ilmenevät tietokoneohjelmien kohdalla ohjelmoijan eri ratkaisuvaihtoehtojen välillä tekemissä valinnoissa, mutta ohjelmoijan persoonallisuuden ei varsinaisesti tarvitse ohjelmasta välittyä<sup>153</sup>. Jos ohjelma edustaa kyseisen tietojenkäsittelyongelman ainoaa järkevää ratkaisua, johon on sitä paitsi päädytty mekaanisesti ja ulkoisten seikkojen pakosta, ei ohjelmaa voida pitää tekijän luovan toiminnan tuloksena<sup>154</sup>. Saarenpään mukaan tällaisessa tilanteessa teoksen idea ja toteutus samastuvat<sup>155</sup>. Standardit, ohjelmointikielet ja tekniset ympäristöt asettavat ohjelmointityölle tietyt reunaehdot, mutta niidenkin rajoissa luovien valintojen tekeminen on usein mahdollista<sup>156</sup>. Tekijänoikeussuojaa ei saa yksinkertainen ohjelma, joka koostuu yleisesti käytetyistä tai alan ammattilaiselle itsestään selvistä ratkaisuksista<sup>157</sup>. Luovaksi toiminnaksi ei ole katsottu sellaista tietokoneohjelman koodaamista, joka on nojautunut pitkälti käyttöohjeeseen<sup>158</sup>.

Tekijänoikeusneuvosto on lausunnossaan TN 2003:10 linjannut, ettei koodiin kirjoitetuilla kommentteilla ole teosarvioinnissa painoarvoa. Lausunnoissa TN 2006:12 ja TN 2013:1 tällaisilla kommentteilla oli kuitenkin merkitystä ohjelmien itsenäisyyden arvioimisessa, sillä sanasta sanaan yhtenevien kommenttien katsottiin viittaavan koodin suoraan kopiointiin aiemmasta ohjelmasta. TN 2003:10 mukaan huomiota ei tule kiinnittää myöskään koodin typografiseen asuun, kuten käskyjen rivitykseen tai pien- ja suuraakkosten käyttämiseen.

Tietokoneohjelmia koskeissa tekijänoikeusneuvoston lausunnoissa mainitaan lähes poikkeuksetta lähdekoodin ohjelmointikieli sekä koodin pituus. Vaikka suomalais-

---

152 Direktiivin mukaan tietokoneohjelmaa suojataan, jos se on tekijänsä henkinen luomus. Muita arviointiperusteita ei ohjelman tekijänoikeudellisen suojattavuuden arvioinnissa saa käyttää. Suomalaisen teostasovaatimuksen kritiikistä ks. Mylly 2005, s. 753–754 sekä Takki 2003, s. 48–49.

153 KM 1987:8, s. 177 ja Välimäki 2009, s. 19

154 HE 161/90 vp., s. 50 ja KM 1987:8, s. 177. Sekä HE:ssä että KM:ssä tällaista tilannetta on kuitenkin pidetty harvinaisena ja katsottu tietojenkäsittelyongelmaan olevan tavallisesti useita ratkaisuvaihtoehtoja.

155 Saarenpää 2001, s. 180

156 Ks. esim. TN 2013:1

157 KM 1987:8, s. 177. Saarenpään mukaan idean ja sen toteutuksen kaksijako lakkaa olemasta ja idea samastuu toteutukseen, kun turvaudutaan vakiintuneesti käytettyihin ratkaisuihin. Ks. Saarenpää 20012, s. 186.

158 Ks. TN 1996:3



sa tekijänoikeudessa suojaa ei myönnetä ns. skill and labour -perusteella, on neuvoston käyttämä asiantuntija toisinaan kommentoinut myös kyseessä olleen ohjelman koodaamisen vaatimaa asiantuntemusta ja työn määrää<sup>159</sup>. Ohjelmistodirektiivin johdanto-osassa kielletään ohjelmiston laadun tai esteettisen arvon tarkasteleminen osana teosarviointia, joten ohjelman ei voida vaatia olevan varsinaisesti laadukas tai korkeaa ammattitaitoa ilmentävä. Myös toimimatonta tai virheellistä ohjelmaa suojataan tekijänoikeudella.

Vaikka ohjelman kaikki osat eivät ilmentäisi tekijän luovaa panosta, voi ohjelma kokonaisuutena kuulua tekijänoikeussuojan piiriin. Esimerkiksi lausunnossa TN 2008:13 ohjelman katsottiin yltävän kokonaisuutena teostasoon, vaikka se sisälsi myös kolmansilta tahoilta hankittuja osia. Jos ohjelmaan on kuitenkin sisällytetty toisen ohjelman tekijänoikeussuojan alaisia osia sellaisenaan tai muunneltuna, voi kyse olla tekijänoikeuden loukkauksesta<sup>160</sup>. Myös alusta eli uuden tietokoneohjelman pohjana käytetty tietokoneohjelma voi saada tekijänoikeussuojaa, jos se yltää teostasoon<sup>161</sup>.

#### **4.3.2 Rajapinnat ja tekijänoikeus**

Vaikka tietokoneohjelman suojaaminen kirjallisena teoksena on ymmärretty ohjelman kielellisen muodon eli koodin suojaamiseksi, ei ohjelmistokehitystä voida tarkastella vain koodin kirjoittamisena. Ohjelmoijan näkökulmasta keskeistä voi olla myös ohjelman rakenteen ja rajapintojen suunnittelu, kun taas ohjelman käyttäjää kiinnostavat lähinnä ohjelman toiminnallisuus ja käyttöliittymä<sup>162</sup>. Ohjelman toiminnallisuus sinänsä ei voi saada tekijänoikeussuojaa, koska se merkitsisi ideoiden suojaamista<sup>163</sup>. Ohjelman rakennettaan ei suojata<sup>164</sup>.

Rajapinta määritellään ohjelmistodirektiivin johdanto-osan 10. perustelukappaleessa ohjelman osaksi, joka mahdollistaa yhteenliittämisen ja vuorovaikutuksen ohjelmiston ja laitteiston osien välillä. Suomessa rajapintojen ei tyypillisesti ole katsottu yltävän teostasoon omaperäisyyden puutteen vuoksi, mutta myös rajapintojen tehtävä yhteentoimivuuden mahdollistajina puoltaa niiden kierrättämisen sallimista<sup>165</sup>. Tietokoneen ja käyttäjän vuorovaikutuksen mahdollistava käyttöliittymä näyttäisi jäävän ohjelmistodirektiivin

---

159 Ks. TN 2003:10 ja TN 2013:1

160 Ks. TN 2013:1

161 Ks. TN 2006:5

162 Välimäki 2009, s. 21 ja Takki 2003, s. 41

163 Ks. EUT C-406/10

164 Takki 2003, s. 63

165 Mylly 2005, s. 766

rajapinnan määritelmän ulkopuolelle<sup>166</sup>, ellei huomiota kiinnitetä samaisen 10. perustelukappaleen alkuosassa viitattuun tietokoneohjelman ja käyttäjän väliseen kommunikaatioon. Euroopan unionin tuomioistuimen (EUT) ratkaisussa C-393/09 (Bezpečnostní Softwarová Asociace) käyttöliittymän onkin tulkittu kuuluvan ohjelmistodirektiivissä mainittuihin rajapintoihin.

Tietokoneohjelman koodi voidaan nähdä myös osana erilaisten elementtien kokonaisuutta, johon sisältyy muitakin teoslajeja. EUT katsoi videopelejä koskeneessa ratkaisussaan C-355/12 (Nintendo), että pelin graafikoilla ja äänillä oli niiden konekielisestä koodimuodosta erillinen ”oma luova arvo, jota ei voida supistaa koodaukseen”. Videopelin koko multimodaalista kokonaisuutta voitiin tuomioistuimen mukaan suojata teoksena tietoyhteiskuntadirektiivin<sup>167</sup> mukaisesti. Samansuuntaisesti tapauksessa EUT C-393/09 (BSA) todettiin, että tietokoneohjelman graafinen käyttöliittymä saattoi saada tekijänoikeussuojaa tietoyhteiskuntadirektiivin perusteella, vaikka käyttöliittymän ei katsottu kuuluvan ohjelmistodirektiivissä säädetyn tietokoneohjelman tekijänoikeussuojan piiriin. Käyttöliittymä kokonaisuudessaan tai siihen kuuluva yksittäinen elementti voi siis olla tekijänoikeuden kohteena, mutta se kuuluu eri teoslajiin (esim. kuvateoksiin) kuin tietokoneohjelmat. Ratkaisussa EUT mainitsi käyttöliittymän teosarvioinnissa huomioitavina piirteinä sen osien erityisen sijoittelun tai suunnittelun. Tuomioistuimen mukaan osat, jotka ovat vain niiden teknisen tehtävän sanelemia, eivät omaperäisyyden puutteen vuoksi ole teoksia.

Käytännössä käyttöliittymien on katsottu vain harvoin yltävän teostasoon<sup>168</sup>. Koska käyttöliittymä ei ole tietokoneohjelman ilmaisumuoto, sitä koskevaa teosarviointia ei suoriteta tietokoneohjelmien alhaiseen teoskynnykseen nojaten. Lausunnossa TN 2016:15 tietokoneen käyttöliittymäpiirustuksia ei pidetty teoksina, sillä käyttöliittymän teknisen tehtävän ja toiminnallisten näkökohtien arvioitiin ratkaisevasti määritelleen käyttöliittymän elementtien ilmaisumuodon. Käyttöliittymän ulkoasua luonnehditaan lausunnossa pelkistetyksi ja selkeäksi, mikä lienee yhteydessä neuvoston johtopäätökseen käyttöliittymän funktionaalisuuden sanelemasta muodosta. Perustelu noudattaa samaa logiikkaa kuin tekijänoikeusneuvoston monet taideteollisia tuotteita koskevat lausunnot, joissa muodon

<sup>166</sup> Näin tulkitsee esim. Mylly 2005, s. 746.

<sup>167</sup> Euroopan parlamentin ja neuvoston direktiivi 2001/29/EY tekijänoikeuden ja lähioikeuksien tiettyjen piirteiden yhdenmukaistamisesta tietoyhteiskunnassa

<sup>168</sup> Takki 2003, s. 63

on katsottu olevan lähinnä seurausta tuotteen käyttötarkoituksesta<sup>169</sup>. Huomiota tulee kuitenkin kiinnittää siihen, ettei käyttöliittymä pelkisty vain graafiseen ulottuvuuteensa. Ohjelman tuottamien kuvien, äänten ja tekstin ohella käyttöliittymää voidaan tarkastella myös käyttäjän ja ohjelman kokonaisvaltaisempana kommunikaatiotapana. Jälkimmäisen tekijänoikeudellinen luonne on kuvia, ääntä ja tekstiä epäselvempi, sillä em. ulottuvuuksilla on selkeämpi liittymä perinteisiin teoslajeihin.<sup>170</sup>

## 4.4 Kokeilutilanne ja teoksen laillinen käyttäminen

### 4.4.1 Taloudelliset ja moraaliset oikeudet

Tekijänoikeus antaa tekijälle lähtökohtaisesti yksinoikeuden luomaansa teokseen. Yksinoikeus rakentuu oikeudesta hyödyntää teosta TekijäL 2 §:n tarkoittamilla tavoilla (positiivinen oikeus) sekä oikeudesta estää muita hyödyntämästä teosta vastaavalla tavalla (negatiivinen eli kielto-oikeus)<sup>171</sup>. Tekijällä ei kuitenkaan ole absoluuttista määräysvaltaa teokseensa, vaan määräysvallan alaa supistavat TekijäL 2 luvussa säädetty tekijänoikeuden rajoitukset.

Tekijän oikeusasema koostuu taloudellisista ja moraalisista oikeuksista. TekijäL 2 § ilmaisee tekijän taloudelliset oikeudet, joihin kuuluvat yksinoikeus kappaleen valmistamiseen teoksesta sekä teoksen saattaminen yleisön saataviin. Saman pykälän 2 momentti määrittelee tarkemmin, millaisia toimia pidetään kappaleen valmistamisena. Kappale voidaan valmistaa kokonaan tai osittain, suoraan tai välillisesti, tilapäisesti tai pysyvästi sekä millä keinolla ja missä muodossa tahansa sekä siirtämällä teos laitteeseen, jolla se voidaan toisintaa. Momentin tulkitaan kattavan kaikenlaiset – myös vasta tulevaisuudessa keksittävät – teoskappaleen valmistusmenetelmät eli tavat antaa teokselle fyysinen muoto<sup>172</sup>. Valmistettujen kappaleiden määrällä tai tarkoituksella ei ole tekijänoikeudellisen arvioinnin kannalta merkitystä<sup>173</sup>.

Tietokoneella tapahtuvaan ohjelmien ja teosten käyttöön sisältyy käytännössä aina kappaleiden valmistamista. Myös tietokonetta tai -verkkoa käytettäessä syntyvät tilapäiset

169 Muotoilutuotteiden teostason arvioinnista ks. esim. TN 2013:20 (vaaput), TN 2013:7 (hana) ja TN 2012:9 (avainpesä ja ovenkahva). Ko. lausuntojen kohteena olevia esineitä ei pidetty teoksina, sillä esineiden käyttötarkoitusten katsottiin määräävän esineiden muodon valintaa.

170 Koivumaa 1995, s. 60–61

171 Pihlajarinne 2014, s. 861

172 Haarmann 2014, s. 72–73

173 Sorvari 2007, s. 71

kopiot on katsottu teoskappaleiksi, joiden syntyminen on siis kappaleen valmistamista.<sup>174</sup> Tietokoneohjelmien kohdalla tekijänoikeuslaki sallii ohjelman käytön edellyttämät kopiot (25 j.1 §), varmuuskopioinnin (25 j.2 §) sekä dekompilaatiokopiot (25 k §). TekijäL 12 §:ää, joka sallii muutaman teoskappaleen valmistamisen yksityiseen käyttöön, ei sovelleta tietokoneohjelmiin, mutta TekijäL 56 a §:n mukaan ohjelman yksityinen kopiointi ei siitä huolimatta ole rangaistavaa.

TekijäL 2.3 §:n mukaan teoksen saattaminen yleisön saataviin voi tapahtua joko yleisölle välittämällä, julkisesti esittämällä, yleisön keskuuteen levittämällä tai julkisesti näyttämällä. Yleisölle välittäminen kattaa teknologianeutraalisti kaiken sellaisen yleisön saataviin saattamisen, jossa teoksen ja yleisön välillä on jonkinlainen maantieteellinen välimatka<sup>175</sup>. Välittäminen pitää 2.3 §:n 1 kohdan mukaan sisällään myös sellaisen teoksen interaktiivisen välittämisen, jossa yleisöön kuuluvilla henkilöillä on mahdollisuus saada teos saataviinsa itse valitsemastaan paikasta itse valitsemanaan aikana (*on demand*). Julkinen esittäminen puolestaan tarkoittaa teoksen välitöntä (esim. live-konsertti) tai välillistä (esim. saman konsertin esittäminen televisiossa) esittämistä esitystapahtumassa fyysisesti läsnä olevalle yleisölle<sup>176</sup>. Yleisön keskuuteen levittäminen voi TekijäL 2.3 §:n 3 kohdan mukaan olla teoskappaleen myymistä, vuokraamista, lainaamista tai muuta levittämistä. Levittämisessä voi olla kyse ainoastaan fyysisen esineen saattamisesta yleisön saataviin<sup>177</sup>. Myös 4 kohdan tarkoittama julkinen näyttäminen voi kohdistua vain fyysiseen teoskappaleeseen, joka asetetaan yleisön nähtäväksi teknisiä apuvälineitä käyttämättä<sup>178</sup>.

2.3 §:n mukainen tekijän yksinoikeus kytkeytyy yleisön käsitteeseen. Komiteanmietinnössä 1953:5 on todettu teoksen yksityisen ja yleisen käyttämisen erottelemisen voivan olla haastavaa, ja asia onkin ratkaistava kussakin yksittäistapauksessa erikseen<sup>179</sup>. Yleisönä eli avoimena piirinä voidaan pitää sellaista joukkoa, johon kuuluvia henkilöitä ei ole yksilöllisesti ennalta määritelty. Esimerkiksi suppeaan ystävä- tai perhepiiriin rajautuva henkilöjoukko ei muodosta yleisöä.<sup>180</sup> Rajanveto avoimen ja suljetun piirin välillä muuttuu haasteelliseksi tilanteissa, joissa osallistujapiiri on jollain kriteerillä rajattu, mutta mah-

<sup>174</sup> Sorvari 2007, s. 62, 76

<sup>175</sup> Harenko ym. 2016, s. 45

<sup>176</sup> Sorvari 2007, s. 110

<sup>177</sup> Sorvari 2007, s. 139. Vrt. Välimäki 2009, s. 41, joka katsoo myös Internetissä tapahtuvan tiedostojen yleisön saataviin saattamisen olevan ratkaisussa KKO 1999:15 otetun kannan mukaisesti kappaleiden levittämistä. Huomattava kuitenkin on, että ratkaisu on annettu ennen kuin 2 §:ään lisättiin yleisölle välittämistä koskeva kohta tekijänoikeuslain muutoksella 821/2005.

<sup>178</sup> Sorvari 2007, s. 155–156

<sup>179</sup> KM 1953:5, s. 47

<sup>180</sup> Haarmann 2014, s. 75 ja Kivimäki 1966, s. 33–34

dollisten osallistujien määrä voi silti kasvaa suureksi. Sorvarin näkemyksen mukaan avoimeksi piiriksi voitaisiin katsoa vasta selvästi 50 henkeä suurempi joukko.<sup>181</sup> Piirin avoimuuden määrittelyn kannalta ei tavallisesti ole merkitystä sillä, onko teoksen saattaminen piiriin kuuluvien henkilöiden saataviin tuloa tuottavaa vai ei<sup>182</sup>.

Edellä kuvattuun avoimeen henkilöpiiriin suuntautuvan toiminnan lisäksi teoksen julkisena esittämisenä ja yleisölle välittämisenä pidetään TekijäL 2.4 § mukaan myös esittämistä ja välittämistä ansiotoiminnassa suurehkolle suljetulle piirille. Kyse on ns. teollisuusmusiikkisäännöksestä, joka on alkujaan säädetty työpaikoilla soitettavaa musiikkia silmällä pitäen, mutta jota voidaan soveltaa muunkinlaisiin teoksiin<sup>183</sup>. Peukalosääntönä suurehkon suljetun piirin määrittelyssä voidaan pitää sitä, että alle 20 henkilön piiri ei koskaan ole suurehko, mutta yli 50 hengen piiri on sitä aina. Lopulta arvio on kuitenkin tehtävä aina kyseessä olevan yksittäistapauksen piirteiden mukaan. Ansiotarkoitukseksi on katsottu kaikki voittoa tavoitteleva toiminta, kunhan teoksen esittämisestä tai välittämisestä vastaa siitä mahdollisesti hyötyvä taho.<sup>184</sup>

Jos kokeilussa testattavaan tietokoneohjelman prototyyppiin on kopioitu ulkopuolisen tahon tuottamaa tekijänoikeuden alaista materiaalia (esim. toisen laatimaa koodia), on kyse TekijäL 2.2 §:n tarkoittamasta kappaleen valmistamisesta. Kokeilun toteuttamistavasta riippuen kyse voi lisäksi olla 2.3 §:n mukaisesta teoksen saattamisesta yleisön saataviin. Esimerkiksi jos ohjelman varhainen versio tarjotaan Internetissä kenen tahansa ladattavaksi, on kyse yleisölle välittämisestä. Toiminta ei kuitenkaan välttämättä suuntaudu 2.3 §:n edellyttämään yleisöön, jos kokeiluun osallistuvien henkilöiden joukko muodostaa suljetun piirin eli jos kokeiluun voi osallistua vain ennalta määriteltä suppeahko joukko. Ulkopuolisten tahojen teosten käyttämistä kokeilussa käsitellään seuraavassa luvussa. Jos kokeilussa syntyy uusia teoksia, kuuluu näidenkin teosten TekijäL 2 §:n mukainen käyttäminen tekijän tai tekijöiden yksinoikeuden alaan. Kokeilussa syntyvien teosten tekijänoikeuksia käsitellään luvussa 6.

Taloudellisten oikeuksien ohella tekijänoikeuteen sisältyvät tekijän persoonaa suojaavat moraaliset oikeudet, jotka määritellään TekijäL 3 §:ssä. 3.1 § mukaan tekijä on hyvän ta-

181 Sorvari 2007, s. 101–102. Tapauksessa I-SHO 7.4.2005, R 03/1060 musiikin ja elokuvien siirtämistä palvelimille ei katsottu avoimeen piiriin suuntautuvaksi, koska palvelimilla oli enimmillään noin 20–30 käyttäjää, eikä käyttäjäksi pääseminen ollut helppoa. Tapauksessa KKO 1999:115 101 käyttäjän joukko katsottiin avoimeksi piiriksi, kun käyttäjäksi vieläpä pääsi helposti.

182 Haarmann 2014, s. 75

183 Sorvari 2007, s. 103

184 Haarmann 2014, s. 83 ja Harenko ym. 2016, s. 65

van mukaisesti ilmoitettava teoskappaleen valmistamisen tai teoksen yleisön saataviin saattamisen yhteydessä (isyysoikeus). 3.2 § mukaan teosta ei ole lupa muuttaa tai saattaa yleisön saataviin tekijää loukkaavalla tavalla (respektioikeus). Isyysoikeudella ei ole ohjelmistoalalla suurta merkitystä, sillä alalla ei juurikaan ole tapana ilmoittaa tekijöiden nimiä tietokoneohjelman yhteydessä. Koska myös respektioikeuden loukkaaminen on ohjelmien kohdalla epätodennäköistä, ei tässä tutkielmassa käsitellä moraalisiin oikeuksiin liittyviä tekijänoikeuskysymyksiä.<sup>185</sup>

#### **4.4.2 Jälkiperaisteokset**

TekijäL 2 §:ssä luetellut tekijän taloudelliset yksinoikeudet koskevat 2.1 §:n mukaan teosta ”muuttamattomana tai muutettuna, käännöksenä tai muunnelmana, toisessa kirjallisuustai taidelajissa taikka toista tekotapaa käyttäen”. Tekijän määräysvalta ei siis rajoitu teokseen vain sen alkuperäisessä muodossa, vaan teosta suojataan muultakin kuin suoralta kopioinnilta.

Tekijänoikeuden alaan kuuluvat teokset voidaan jakaa alkuperäis- ja jälkiperaisteoksiin. Jälkimmäinen ryhmä käsittää käännökset, muut muunnelmat ja kokoomateokset.<sup>186</sup> Jälkiperaisteoksen tekijä saa tekijänoikeuden omaan luovaan panokseensa, mutta hän voi hyödyntää teostaan TekijäL 2 §:n tarkoittamilla tavoilla vain alkuperäisteoksen tekijän luvalla. Näin ollen jälkiperaisteos ei vaikuta alkuperäisteoksen tekijän oikeuksiin, ja jälkiperaisteoksen tekijän oikeudet ovat alkuperäisteoksen tekijästä riippuvaisia.<sup>187</sup> Kolmas henkilö tarvitsee jälkiperaisteoksen hyödyntämiseen sekä alkuperäis- että jälkiperaisteoksen tekijän luvan<sup>188</sup>.

Teoksen muuntelemisesta ja muuttamisesta säädetään TekijäL 4 §:ssä. Muunnelmista voidaan erottaa epäitsenäiset eli varsinaiset muunnelmat ja itsenäiset eli vapaat muunnelmat<sup>189</sup>. Epäitsenäisen muunnelman tekijä ei voi määrätä teoksestaan alkuperäisteoksen tekijänoikeutta loukkaavalla tavalla (TekijäL 4.1 §), kun taas vapaa muunnelma on itsenäinen teos, josta tekijä voi määrätä alkuperäisteoksen tekijänoikeudesta riippumattomasti (TekijäL 4.2 §). 4.2 §:n tarkoittamassa vapaassa muuttamisessa on kyse sallitusta toisten

<sup>185</sup> Pitkänen ym. 2003, s. 81. Kuitenkin erityisesti avoimen lähdekoodin ohjelmissa tekijöiden mainitseminen kuuluu tapoihin, ja tekijöiden nimien liittäminen ohjelmaan voi riskien hallitsemisen kannalta olla muissakin tapauksissa järkevää. Ks. Välimäki 2009, s. 32–33.

<sup>186</sup> Haarmann – Mansala 2012, s. 90 ja Pihlajarinne 2014, s. 881

<sup>187</sup> Harenko ym. 2016, s. 30–31

<sup>188</sup> KM 1953:5, s. 49

<sup>189</sup> Pihlajarinne 2014, s. 881

teoksista inspiroitumisesta ja ideoiden hyödyntämisestä<sup>190</sup>. Jos muuntelijan työn tulos ei ylitä teoskynnystä, ei hänelle tietenkään synny tekijänoikeutta suorittamaansa muunte-  
luun<sup>191</sup>.

Muuntelemisella tarkoitetaan kaikenlaisia muuntelutapoja, jotka jollain tavalla hyödyntä-  
vät alkuperäisteosta tai sen osaa<sup>192</sup>. Tietokoneohjelmien tapauksessa muuntelua tapahtuu  
esimerkiksi silloin, kun ohjelmaa kehitetään tekemällä muutoksia sen lähdekoodiin tai  
kun ohjelma sovitetaan uuteen käyttöympäristöön tai uudelle laitteistolle. Alkuperäiste-  
oksen tekijän määräysvalta teokseen ei katkea silloinkaan, kun tietokoneohjelma käänne-  
tään toiselle ohjelmointikielelle.<sup>193</sup>

Toinen jälkipäätteiksi koskeva keskeinen lainkohta on TekijäL 5 §. Sen mukaan hen-  
kilöllä, joka on muiden tekijöiden (avustajien) teoksia tai teosten osia yhdistelemällä ai-  
kaansaanut kirjallisen tai taiteellisen kokoomateoksen, on kokoomateokseen tekijänoike-  
us. Kokoomateoksen tekijän (toimittajan) oikeus ei kuitenkaan rajoita avustajien oikeutta  
omiin teoksiinsa, vaan toimittajan oikeus käsittää vain sen luovan työn tuloksen, jonka  
hän on kooten, valikoiden ja muuten järjestelemällä saanut aikaan.<sup>194</sup> Ohjelmistokehityk-  
sen kontekstissa kokoomateos voisi tarkoittaa esimerkiksi ohjelmaa, joka on luotu valmi-  
ta koodinpätkiä yhdistelemällä<sup>195</sup>.

Kokeilun yhteydessä tapahtuva prototyypin valmistaminen ja muokkaaminen voi tekijän-  
oikeudellisessa mielessä merkitä myös muunnelman tai kokoomateoksen luomista. Jälki-  
päätteoksen luominen voi liittyä joko prototyypin kokoamiseen olemassa olevaa (jonkin  
kolmannen tahon luomaa) materiaalia muokkaamalla ja yhdistelemällä tai prototyypin  
muokkaamiseen kehitysprosessin aikana eri tahojen toimesta.

---

190 Haarmann 2014, s. 65

191 Välimäki 2009, s. 37, nimittää tällaista toimintaa vähäpätöiseksi muunteluksi ja mainitsee esimerkkinä yksin-  
kertaisten virheiden korjaamisen tietokoneohjelman lähdekoodista.

192 Harenko ym. 2016, s. 32

193 Välimäki 2009, s. 38–39

194 Kivimäki 1966, s. 49–50

195 Innanen – Jäske 2014, s. 148

## 5. NOPEA JA EDULLINEN PROTOTYPOINTI TEKIJÄNOIKEUDELLEEN HAASTEENA

### 5.1 Valmiin materiaalin lisensointi

Kokeillen kehitettäessä olennaista on löytää yksinkertainen, nopea ja edullinen tapa viedä iteraatio läpi<sup>196</sup>. Tekijänoikeusintensiivisillä aloilla, kuten ohjelmistoteollisuudessa, nopean ja edullisen prototypoinnin vaatimus saattaa kuitenkin vaikuttaa osittain haasteelliseltakin ihanteelta, koska juuri tuotekehitys voi olla erittäin kallista ja aikaavievää. Ohjelmistoalalla materiaalikulut ovat tyypillisesti alhaisia ja suurin osa kuluista koostuu yrityksessä tehtävän työn ja ostopalvelujen suorista ja epäsuorista kuluista<sup>197</sup>. Näin ollen juuri tuotekehitykseen käytettävien työtuntien vähentäminen vaikuttaisi olevan keskeinen keino kokeiluprosessin keventämiseen.

Immateriaalioikeuksien osalta yrityksen on valittava, kehittääkö se itse (tai mahdollisesti yhteistyössä jonkin ulkopuolisen tahon kanssa) immateriaalioikeuksien alaisen omaisuutensa vai käyttääkö se jo olemassa olevaa, jonkun toisen luomaa ja omistamaa materiaalia<sup>198</sup>. Jos yritys ei itse tuota kaikkea tekijänoikeuden alaista materiaaliaan, se voi osana tuotekehitystään käyttää laillisesti myös public domainiin kuuluvia teoksia sekä teoksia, joihin on hankittu oikeus joko tekijänoikeuksien siirto- tai lisensointitoimella<sup>199</sup>. Tavallista on hankkia toisen oikeudenhaltijan teokseen vain käyttöoikeus eli lisenssi, koska lisensointi on usein halvempaa kuin oikeuksien kokonaisluovutus tai vastaavan ratkaisun kehittäminen omin neuvoin<sup>200</sup>.

Kun tekijänoikeuden haltija myöntää lisenssin, hän antaa lisenssinsaajalle käyttöluvan johonkin yksinoikeutensa piiriin kuuluvaan kohteeseen, jonka luvatta tapahtuva hyödyntäminen merkitsisi tekijänoikeuden loukkausta<sup>201</sup>. Useinkaan annettava käyttöoikeus ei ole laaja-alainen, vaan se on ajallisesti, alueellisesti ja asiallisesti rajattu – puhutaankin oikeuden ”viipaloinnista ja annostelusta”<sup>202</sup>. Rosenin mukaan lisenssi voi olla joko uni- tai bilateraalin sopimus tai pelkkä lupa. Jos lisenssi hahmotetaan pelkkänä lupana, jonka kohdalla sopimuksen syntymisen edellytykset eivät siis ole toteutuneet, voidaan pitää jossain määrin kyseenalaisena, onko

---

196 Poskela ym. 2015, s. 14

197 Tyrväinen 2003, s. 23

198 Palfrey 2012, s. 7

199 Gollin 2008, s. 154, 268

200 Innanen – Jäske 2014, s. 194

201 Edelman 2012, s. 1098

202 Innanen – Jäske 2014, s. 194



lisenssi täytäntöönpantavissa. Tässä suhteessa lisenssin oikeudellinen luonne voi olla epäselvä esimerkiksi vapaasti Internetissä saatavilla olevien avoimen lähdekoodin ohjelmien kohdalla.<sup>203</sup>

Lisensiointi voi nopeuttaa kehitettävän tuotteen pääsyä markkinoille, koska tällöin jonkun toisen aiemmin kehittämiä ominaisuuksia ei tarvitse alusta asti toteuttaa itse. Lisenssinsaajan ei siis tarvitse investoida ylimääräisiin tutkimus- ja tuotekehitystoimiin vain keksiäkseen pyörän uudestaan.<sup>204</sup> Oikeuksien lisensoinnin tulisi kuitenkin olla harkittua, koska oikeuksista maksetaan täysin turhaan, jos niitä ei lopulta käytetäkään. Toisaalta liian suppea oikeuksien hankinta voi merkitä toisen tekijänoikeuden loukkausta.<sup>205</sup> Myös tekijänoikeuden siirron tai lisensoimisen mahdollistamisen vaatimat kulut eli transaktiokustannukset voivat nousta suuremmiksi kuin tekijänoikeuksia käyttämällä saatavat tulot, jos tekijänoikeuden haltijan jäljittäminen ja hänen kanssaan neuvottelemine vievät paljon resursseja<sup>206</sup>. Jos lisenssin hankkiminen käy liian kalliiksi, voi vastaavan funktion toteuttavan teoksen luominen itse olla kannattavampaa.

Luvussa 4.1 viitatus tekijänoikeuden laajenemisen eräänlaisena vastareaktionä on syntynyt aloitteita, joiden tavoitteena on kasvattaa yhteisesti käytettävissä olevan materiaalin (*commons*<sup>207</sup>) määrää. Tunnetuimpia esimerkkejä tällaisista yhteisresursseista ovat avoimen lähdekoodin ohjelmistoprojektit sekä Creative Commons. Avoimen lähdekoodin lisenssien sekä Creative Commons -lenssien kaltaisten julkisten lisenssien<sup>208</sup> käyttäminen mahdollistaa matalan kynnyksen lisensoinnin ilman korkeita transaktiokustannuksia<sup>209</sup>, joten tällä tavalla lisensoidun valmiin materiaalin käyttäminen voi olla mielekästä myös kokeilussa, jossa resursseja ei vielä ole järkevää käyttää mittavaan lisensointiin.

On syytä korostaa, ettei kokeilun kustannustehokkuus ole riippuvainen vain prototyypistä ja siihen liittyvistä immateriaalioikeuksista, vaan syklien läpiviennin vaatiman ajan, vään ja rahan määrä on seurausta monen eri tekijän yhteisvaikutuksesta. Tällaisia tekijöitä ovat muun muassa kommunikaation sujuvuus, hyödyllisen asiakaspalautteen saatavuus ja kehitystiimin kokoonpano<sup>210</sup>. Resursseja voidaan säästää myös toteuttamalla prototyyppiin

203 Rosen 2004, s. 53–57

204 Vedenkannas 2009, s. 965

205 Ashorn 2003, s. 86 ja Mansala 2007, s. 598

206 Goldstein – Hugenholtz 2013, s. 275 ja Pihlajarinne 2014, s. 878

207 Lessig 2001, s. 19–20, määrittelee commonsin resurssiksi, johon kenellä tahansa tietyn yhteisön jäsenellä on oikeus lupaa pyytämättä. Joissain tapauksissa lupaa voidaan edellyttää, mutta tällöinkin lupa on myönnettävä puolueettomasti.

208 Hietanen 2008, s. 16, määrittelee julkiset lisenssit lisenkseiksi, jotka on ennalta määrättyjen yksilöiden sijasta suunnattu yleisölle, ts. kaikille, jotka lisenssin haluavat.

209 Hietanen 2008, s. 16

210 Nguyen-Duc – Wang – Abrahamsson 2017, s. 32–33

vain kokeilussa testattavien hypoteesien kannalta välttämättömät ominaisuudet. Tässä luvussa keskitytään kuitenkin prototyypin luomisen tekijänoikeudelliseen ulottuvuuteen.

## 5.2 Prototyyppi ilman koodaamista

Tuotekehitysprosessin varhaisemman ja myöhäisemmän vaiheen prototyypit ovat harvoin aivan samantyyppisiä, koska niiden funktiokin on erilainen. Alkuvaiheen prototyypit (*throwaway prototypes*) ovat luonteeltaan kertakäyttöisiä, helposti rakennettavia apuvälineitä, joita ei oteta osaksi valmista tuotetta. Sen sijaan myöhäisemmän vaiheen prototyypit (*evolutionary prototypes*) ovat hienostuneempia ja enemmän resursseja vaativia tuotoksia, jotka luovat pohjan oikealle tuotteelle. Ohjelmistokehityksen alkuvaiheen prototyyppi voi tapahtua vaikkapa paperilla, kun taas pitemmälle viety prototyyppi sisältää jo runsaan määrän koodia.<sup>211</sup>

Varhaiset prototyypit, joilla testataan ideaa, käyttöliittymää tai yksittäisiä toimintoja, eivät välttämättä vaadi lainkaan lähdekoodin kirjoittamista. Kehitettävän idean kiinnostavuutta voidaan kokeilla esimerkiksi kertomalla suunnitteilla olevasta tuotteesta yksinkertaisella verkkosivulla tai videolla, joiden avulla kerätään palautetta potentiaalisilta käyttäjiltä. Tyypillistä on myös luoda nopeita prototyyppiejä erilaisten online-työkalujen avulla, jolloin visuaalisia ja interaktiivisia malleja voidaan toteuttaa ilman koodausta tai muutakaan teknistä erityisosaamista.<sup>212</sup> Kun prototyyppi toteutetaan ilman lähdekoodia, ei kyse ole vielä tekijänoikeuslain tarkoittamasta tietokoneohjelmasta. Jostain toisenlaisesta teostyyppistä voi toki olla kyse tällaisessakin prototyyppinnissa. Kovin pelkistetty rautalankamallituskin ylittäisi teoskynnystä, mutta esimerkiksi pidemmälle viety graafisen käyttöliittymän mockup saattaisi saada suojaa kuvateoksena.

## 5.3 Komponenttipohjainen ohjelmointi

Laajoja ohjelmistoja luodaan harvoin tyhjästä, vaan kehityksessä hyödynnetään uudelleen käytettäviä ohjelmistoyksiköjä eli komponentteja, joita yhdistelemällä saadaan aikaan

<sup>211</sup> Nguyen-Duc ym. 2017, s. 22, 25

<sup>212</sup> Hassi ym. 2015, s. 150–156 ja Nguyen-Duc ym. 2017, s. 27. Online-työkaluja, joita voidaan käyttää ohjelman ja käyttäjän välistä interaktiota havainnollistavien prototyyppien toteuttamisessa, ovat esimerkiksi InVision ([invisionapp.com](https://invisionapp.com)), Justinmind ([justinmind.com](https://justinmind.com)) ja Balsamiq ([balsamiq.com](https://balsamiq.com)).

suurempia kokonaisuuksia. Valmiiden osien hankkiminen on mielekästä tapauksissa, joissa aikaa, valmiuksia tai resursseja ei ole jonkin toiminnon itsenäiseen kehittämiseen.<sup>213</sup>

Komponentteja voidaan hankkia eri tavoin. Yrityksen itse kehittämiä komponentteja voidaan käyttää uudelleen, kaupallisia komponentteja voidaan ostaa toiselta yritykseltä (*components off-the-shelf* eli COTS), komponentteja saatetaan hankkia Open Source -yhteisöltä tai komponenttien kehitys on ehkä ulkoistettu.<sup>214</sup> Kehitettävä systeemi voidaan koota sel-  
laisistakin osista, jotka on alkujaan luotu aivan muihin käyttötarkoituksiin ja joita ei ole tarkoitettu toimimaan yhdessä. Valmiita komponentteja käytettäessä aikaa kuluu vähemmän erilaisten toimintojen kehittämiseen, mutta komponenttien liittäminen systeemiin ei välttämättä ole mutkatonta.<sup>215</sup>

Kaupallisten komponenttien eräs pääasiallinen heikkous on suljettu lähdekoodi. Kun komponentin lähdekoodiin ei päästä käsiksi, se joudutaan integroimaan systeemiin mustana laatikkona, jonka sisältöä ei voida muokata tai kunnolla edes tuntea.<sup>216</sup> Kun komponentin lähdekoodi on ohjelmiston kehittäjille mysteeri, on ohjelmistossa havaittujen virheiden paikantaminenkin vaikeaa. Lähdekoodin suljetuudesta johtuvat ongelmat voidaan välttää käyttämällä avoimen lähdekoodin komponentteja. Avointa lähdekoodia hyödynnetään käytännössä varsin yleisesti ohjelmistokehityksessä, koska parhaimmillaan avoimen lähdekoodin lisensointi mahdollistaa muokattavissa olevien ja edistyneiden toimintojen sisällyttämisen omaan ohjelmistoon ilmaiseksi.<sup>217</sup>

## 5.4 Avoimen lähdekoodin hyödyntäminen

### 5.4.1 Mitä avoin lähdekoodi on?

Tyypillisesti käyttäjän ei ole mahdollista päästä käsiksi kaupallisten tietokoneohjelmien lähdekoodiin, sillä tietokoneohjelman oikeudenhaltijalla on usein liiketoimintansa kannalta vahva intressi salata lähdekoodi. Näin ei kuitenkaan aina ole. Avoimen lähdekoodin ohjelmistoilla (*Open Source Software, OSS*) viitataan sellaisiin ohjelmistoihin, joiden lähdekoodia voi kuka tahansa vapaasti tarkastella, muunnella, kopioida ja levittää – kunhan

213 Di Giacomo 2005, s. 301 ja German – Hassan 2009, s. 188

214 Badampudi – Wohlin – Petersen 2016, s. 105

215 Ncube – Oberndorf – Kark 2008, s. 38–39

216 Di Giacomo 2005, s. 301

217 Li – Conradi – Slynstad – Bunse – Torchiano – Morisio 2006, s. 899, Ncube ym. 2008, s. 40 ja Nguyen-Duc ym. 2017, s. 30

noudattaa lisenssin ehtoja<sup>218</sup>. Avoimen lähdekoodin määrittelemiseen käytetään melko vakiintuneesti Open Source Initiativen (OSI)<sup>219</sup> laatimaa määritelmää (Open Source Definition, OSD), jota OSI käyttää kriteerinä sertifioidessaan lisenssejä avoimen lähdekoodin lisensseiksi. Ohjelma on ”aidosti” avointa lähdekoodia, jos se on lisensoitu OSD:n ehtojen mukaisella sertifioidulla lisenssillä.<sup>220</sup>

OSD pitää sisällään 10 kohtaa, joista keskeisimpinä voidaan pitää kohtia 1–3<sup>221</sup>. Ensimmäisen kohdan mukaan ohjelman tulee olla vapaasti levitettävissä ja välitettävissä. Toisen kohdan mukaan lähdekoodi on toimitettava ohjelman mukana tai sen on oltava muuten vapaasti saatavilla. Kolmas kohta koskee ohjelmasta johdettuja teoksia (*derivative work*), joiden luominen ja levitys tulee olla sallittua. Muissa kohdissa muun muassa kielletään erilaisten henkilöiden ja ryhmien eriarvoinen kohtelu sekä käyttötarkoitusten rajoittaminen.<sup>222</sup>

Avoimen lähdekoodin määritelmän täyttävät lisenssiehdot kannustavat käyttämään ohjelmiston osia uudelleen uusissa systeemeissä joko komponentteina tai lähdekoodina, ja avoimen lähdekoodin suosion kasvaessa komponenttien kierrätys on muuttunut tavalliseksi käytännöksi<sup>223</sup>. Myös GitHubin kaltaiset yhteisölliset OSS-ohjelmointialustat ovat olleet omiaan normalisoimaan koodin lainaamisen toisista projekteista<sup>224</sup>. Siinä missä suljetun lähdekoodin komponenttia joudutaan yleensä käyttämään toimittajan rajaamana tuotteena, on avointa lähdekoodia mahdollista omaksua esimerkiksi muutaman koodirivin verran tai käyttää uudelleen kokonaista luokkaa, kirjastoa, komponenttia tai systeemiä<sup>225</sup>.

Avoimen lähdekoodin piiristä voidaan erottaa OSI:n edustaman Open Source -yhteisön lisäksi myös Free Softwarena tunnettu yhteisö. Molemmat ryhmät pyrkivät avaamaan ohjelmistotuotteita laajemmille ihmisjoukoille ja voimaannuttamaan käyttäjiä lähdekoodin avoimuuden kautta, mutta ryhmillä on myös tiettyjä painotuksellisia eroja. Free Software keskittyy enemmän ohjelmistoteollisuuden muuttamiseen ideologisella tasolla, kun taas Open Source pyrkii osallistamaan avoimen lähdekoodin projekteihin erilaisia toimijoi-

218 Välimäki 2002, s. 851 ja Wu – Manabe – Kanda – German – Inoue 2017, s. 1195

219 Open Source Initiative on voittoa tavoittelematon järjestö, joka hallinnoi avoimen lähdekoodin määritelmää, arvioi lisenssejä määritelmää vasten ja myöntää sertifikaatin määritelmän täyttävälle lisensseille. ”About the Open Source Initiative”. Open Source Initiative. <https://opensource.org/about>. 24.11.2017.

220 Dixon 2004, s. 8

221 Välimäki 2009, s. 204

222 ”Open Source Definition”. Open Source Initiative. <https://opensource.org/osd>. 24.11.2017.

223 German – Di Penta – Gueheneuc – Antoniol 2009, s. 82 ja Wu ym. 2017, s. 1195

224 Gharehyazie – Ray – Filkov 2017, s. 291

225 Spinellis – Szyperski 2004, s. 29

ta – kuten tärkeitä kaupallisia ohjelmistokehittäjiä – ilman ideologista sitoutumista avoimuuteen.<sup>226</sup> Avoimesta lähdekoodista tai Open Sourcesta puhutaan usein silloinkin, kun viitataan samanaikaisesti sekä Open Source Softwareen että Free Softwareen<sup>227</sup>. Tällainen laajempi määritelmä on omaksuttu myös tässä tutkielmassa.

Free Software Foundation (FSF)<sup>228</sup> määrittelee vapaan ohjelman neljän perusvapauden avulla: vapaata ohjelmaa saa käyttää kaikenlaisiin tarkoituksiin, ohjelmaa saa tutkia ja muuttaa, ohjelman kopioita saa levittää lähimmäisen auttamiseksi ja ohjelman modifioituja versioita saa levittää ja siten antaa koko yhteisölle mahdollisuuden hyötyä tehdyistä muutoksista. Perusvapauksien täyttyminen edellyttää ohjelman lähdekoodin saatavilla pitämistä.<sup>229</sup> FSF:n määritelmä on siis ydinosaan varsin samankaltainen kuin OSI:n määritelmä, joskin FSF:n määritelmän sanavalinnoissa korostuvat vapaus ja yhteisölliset tavoitteet.

Vaikka OSI:n ja FSF:n määritelmät asettavat tietyt reunaehdot avoimen lähdekoodin lisenssien sisällölle, eivät eri lisenssit ole keskenään identtisiä. Lisenssejä voidaankin ryhmitellä eri tavoin. Varsin yleinen tapa on jakaa lisenssit karkeasti kahteen ryhmään: salliviin lisensseihin (*permissive licenses*) ja vastavuoroisuutta vaativiin lisensseihin (*reciprocal licenses*).<sup>230</sup> Jako perustuu siihen, millaisia ehtoja lisenssissä asetetaan koodin muokkaamisen ja johdettujen teosten luomisen suhteen.<sup>231</sup> Lisenssien erojen ymmärtäminen on keskeistä hyödynnettäessä avointa lähdekoodia osana kehitettävää systeemiä.

#### 5.4.2 Sallivat lisenssit

Salliviksi lisensseiksi kutsutaan avoimen lähdekoodin lisenssejä, jotka asettavat verrattain vähän rajoitteita lisensioidun ohjelman hyödyntämiselle. Ne mahdollistavat ohjelman vapaan levittämisen, kopioimisen, muuntelemisen sekä johdetun teoksen levittämisen eri lisenssin alaisena. Suosittuja sallivia lisenssejä ovat mm. BSD License, MIT License, Artistic

226 Dixon 2004, s. 2–3, 19–20 ja Lessig 2001, s. 68

227 Rosen 2004, s. 3. Open Source Softwaresta ja Free Softwaresta voidaan käyttää myös molemmat kattavaa lyhennettä "FOSS".

228 Free Software Foundation on voittoa tavoittelematon järjestö, joka ilmoittaa missiökseen tietokoneiden käyttäjien vapauden edistämisen ja ohjelmistojen käyttäjien oikeuksien puolustamisen. FSF tukee vapaiden ohjelmistojen kehitystä ja käyttöä erityisesti GNU-projektin muodossa. "The Free Software Foundation (FSF) is a nonprofit with a worldwide mission to promote computer user freedom." Free Software Foundation. <http://www.fsf.org/about/>. 24.11.2017.

229 "What is Free Software?" Free Software Foundation. <https://www.gnu.org/philosophy/free-sw.html>. 24.11.2017

230 Ks. esim. German ym. 2009, s. 82 ja Välimäki 2009, s. 205–206

231 Välimäki 2005, s. 117. Välimäki mainitsee vaihtoehtoisena lisenssien luokittelutapana lisenssien historialliseen alkuperään perustuvan luokittelun.

License sekä Apache License.<sup>232</sup> Tiettyjä sallivia lisenssejä, kuten BSD- ja MIT-lisenssejä, kutsutaan myös akateemiseksi lisensseiksi, sillä ne on alkujaan luotu akateemisten instituutioiden käyttöön ja ne ilmentävät akateemista vapautta.<sup>233</sup>

Pelkistetyin tavallisimmista avoimen lähdekoodin lisensseistä on Massachusetts Institute of Technologyn käyttöön ottama MIT-lisenssi<sup>234</sup>. Se on myös varsin suosittu: esimerkiksi vuonna 2015 lähes 45% GitHubin projekteista oli MIT-lisensioituja<sup>235</sup>. Lisenssi asettaa lisenssinsaajalle varsin vähän rajoituksia. Lisenssinsaaja saa vapaasti käyttää, muokata, kopioida, julkaista, levittää, sisällyttää toiseen teokseen ja alilisensioida eteenpäin MIT-lisensioitua ohjelmaa sekä myydä ohjelman kopioita<sup>236</sup>. Kuitenkaan lisenssitekstiä, tekijöiden nimiä tai vastuuvapauslauseketta ei saa poistaa ohjelman yhteydestä silloinkaan, kun MIT-lisensioitua ohjelmaa käytetään laajemman systeemin komponenttina<sup>237</sup>.

Kalifornian yliopiston piirissä alkunsa saanut BSD-lisenssi ei juuri eroa MIT-lisenssistä. Erona MIT-lisenssiin alkuperäinen BSD-lisenssi sisälsi kiellon käyttää tekijänoikeuden haltijan tai tekijöiden nimeä johdettujen teosten markkinoinnissa, mutta kyseinen kohta poistettiin lisenssin seuraavasta versiosta.<sup>238</sup> Myös Apache-lisenssi antaa lisenssinsaajalle pitkälti samoja oikeuksia kuin MIT- ja BSD-lisenssit, mutta näistä eroten lisenssi pyrkii myös suojelemaan Apachen tavaramerkkiä. Artistic License painottaa tekijän moraalisia oikeuksia.<sup>239</sup>

Koska sallivat lisenssit eivät kiellä johdettujen teosten uudelleen lisensioimista, on tällaisen lisenssin nojalla levitettyä lähdekoodia mahdollista sisällyttää myös omisteisen lisenssin (*proprietary license*) alla levitettäviin ohjelmiin<sup>240</sup>. Johdettujen teosten lähdekoodi voidaan liittää osaksi kaupallista, vain konekielisessä muodossa levitettävää ohjelmistoa, eikä koodin muuntelijan tarvitse julkaista tekemiensä parannusten lähdekoodia. Koodin kopioimista suljetun lähdekoodin tuotteisiin ja projektin mahdollista haarautumista eri suuntiin ei pidetä OS-yhteisössä erityisen toivottuina ilmiöinä, mutta toisaalta sallivien lisenssien käyttäminen voi houkutella projekteihin mukaan myös sellaisia kaupallisia toimijoita, jotka karttavat vastavuoroisuutta edellyttäviä projekteja<sup>241</sup>.

232 German ym. 2009, s. 82 ja Välimäki 2005, s. 118

233 Rosen 2004, s. 69

234 Välimäki 2009, s. 208

235 Balter 2015

236 "The MIT License". Open Source Initiative. <https://opensource.org/licenses/MIT>. 25.11.2017.

237 Välimäki 2009, s. 208

238 Rosen 2004, s. 74 ja Välimäki 2009, s. 208–209

239 Rosen 2004, s. 92, 95

240 Rosen 2004, s. 70

241 Dixon 2004, s. 47–48

### 5.4.3 Vastavuoroisuutta vaativat lisenssit

Vastavuoroisuutta vaativat lisenssit voidaan ryhmitellä sen perusteella, miten voimakkaan vastavuoroisuusvelvoitteen lisenssi asettaa. Heikon vastavuoroisuusehdon (*standard reciprocity obligation*) sisältävä eli pysyvä lisenssi edellyttää, että lisensoitu ohjelma ja siihen tehdyt muutokset pysyvät alkuperäisen lisenssin alaisuudessa. Lähdekoodia ei siis muunneltunakaan voida sulkea. Jos pysyvän lisenssin alla levitetty komponentti kuitenkin yhdistetään esimerkiksi linkittämällä osaksi laajempaa teoskokonaisuutta, ei kokonaisuutta tarvitse lisensoida komponentin alkuperäisin lisenssiehdoin. Voimakkaan vastavuoroisuusehdon (*strong reciprocity obligation*) asettavat eli tarttuvat lisenssit puolestaan aiheuttavat ns. virusvaikutuksen (*viral effect*). Kun tarttuvan lisenssin alaista lähdekoodia yhdistetään muunlaisen lisenssin alaiseen ohjelmistoon, laajenee tarttuva lisenssi kattamaan koko teoksen. Vastavuoroisuutta kutsutaan toisinaan myös hieman leikkillisesti ”copyleftiksi”<sup>242</sup>. Yleisesti käytettyjä pysyviä lisenssejä ovat mm. GNU Lesser General Public License (GNU LGPL) ja Mozilla Public License (MPL), kun taas tarttuvista lisensseistä tunnetuin esimerkki on GNU General Public License (GNU GPL) eri versioineen.<sup>243</sup> Koska GPL on laajimmin käytetty vastavuoroisuutta vaativa lisenssi<sup>244</sup>, koskee suurin osa vastavuoroisuutta vaativiin lisensseihin liittyvästä keskustelusta erityisesti GPL:ä.

Aina ei ole täysin selvää, koskeeko vastavuoroisuusehto koko teosta silloin, kun vastavuoroisuutta vaativan lisenssin alainen komponentti liitetään uuteen teokseen. Vastavuoroisuuden vaatimus kiinnittyy yleensä teoksen levittämiseen sekä johdetun teoksen käsitteeseen, joiden tulkinta on avainasemassa ratkaistaessa, voidaanko avointa lähdekoodia hyödyntää ilman, että myös omaa lähdekoodia täytyisi levittää saman lisenssin alaisena. Esimerkiksi GPL koskee vain julkaistuja ja edelleen levitettäviä teoksia, jolloin pelkkä ohjelman yksityinen käyttäminen ja modifioiminen eivät velvoita käyttäjää noudattamaan lisenssiehtoja. Itsestään selvää ei toki välttämättä ole, millainen toimi merkitsee lisenssissä tarkoitettua julkaisemista ja levittämistä.<sup>245</sup>

Vastavuoroisuutta vaativien lisenssien kohdalla johdetun teoksen käsitteen määrittely on keskeistä, sillä vastavuoroisuus edellyttää johdettujen teosten tuleamista saman lisens-

242 Rosen 2004, s. 105. Nimitys viittaa vastavuoroisuutta vaativien avoimen lähdekoodin lisenssien vaikutukseen, joka on tavallaan päinvastainen tekijänoikeuslain antamiin yksinoikeuksiin nähden: sen sijaan, että tekijä käyttäisi yksinoikeuksiaan, hän lisensoimalla kasvattaa yhteiskäyttöisten resurssien määrää.

243 Välimäki 2005, s. 118 ja Välimäki 2009, s. 206

244 Balter 2015 ja ”Top Open Source Licenses”. Black Duck Software. <https://www.blackducksoftware.com/top-open-source-licenses>. 1.12.2017. Balterin esittämän tilaston mukaan vuonna 2015 12.96 % GitHubin projekteista käytti GPL 2.0 -lisenssiä ja 8.88 % GPL 3.0 -lisenssiä. Black Duckin tilastoissa 18 % projekteista käytti GPL 2.0 -lisenssiä ja 7 % GPL 3.0 -lisenssiä.

245 Välimäki 2005, s. 132, 137–138 ja Välimäki 2009, s. 212



sin alaisuuteen. Näin ollen esimerkiksi GPL-lisensioidusta teoksesta johdettu teos tulee GPL:n alaisuuteen, kun johdettua teosta levitetään. Johdetun teoksen käsitteen tulkinta on ongelmallista, sillä Yhdysvalloissa laaditut avoimen lähdekoodin lisenssit liittyvät sikäläiseen oikeusjärjestelmään omaleimaisine käsitteineen. ”Derivative work” määritellään Yhdysvaltain Copyright Actissa seuraavasti:

*A “derivative work” is a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications, which, as a whole, represent an original work of authorship, is a “derivative work”. (17 U.S.C. § 101.)*

Lainkohdasta ilmenee, että määritelmä pitää sisällään laajan kirjon jälkiperäisteoksia, joiden tarkkarajaiseen määrittelemiseen ei ole laissa edes pyritty, vaan lainkohdan luettelo on esimerkinomainen. Yhdysvaltain lainsäädännön näkökulmasta johdettu teos on jotain, mikä sijoittuu olemassa olevia teoksia muuttamattomana yhdistelevän kokoomateoksen ja uuden itsenäisen teoksen välimaastoon. Johdettu teos koostuu olennaisilta osin jo aiemmin luodusta materiaalista, mutta se ilmentää myös uudenlaista luovaa panosta, joka on toteutettu muutoksina alkuperäiseen materiaaliin.<sup>246</sup>

Jos Copyright Actin määritelmä ylipäänsä onkin laava ja tulkinnanvarainen, on määritelmän katsottu olevan erityisen epäselvä juuri tietokoneohjelmien kohdalla. Useista yhdessä toimivista ohjelmista ja tiedostoista koostuvia ohjelmistoja on vaikea tarkastella suhteessa määritelmään, eikä tulkintaongelmia ole juuri selvennetty oikeuskäytännössä.<sup>247</sup> Tietokoneohjelman yhdistämistä toiseen on oikeuskirjallisuudessa pidetty johdettuna teoksena tiettyjen ehtojen täyttyessä: yhdistelmän tulee olla suhteellisen pysyvä, sen on sisällettävä merkittäviä ja luovia osuuksia toisen ohjelmasta, yhdistelmän on kokonaisuudessaan oltava luova ja sen on sisällettävä merkittäviä ja luovia muutoksia toiseen ohjelmaan. Jos ehdot eivät täyty, on kyse muunlaisesta tekijänoikeuden alaisen materiaalin järjestämisestä, kuten kokoomateoksen luomisesta.<sup>248</sup>

Toisaalta ei välttämättä ole täysin selvää, onko lisenssissä omaksuttu johdetun teoksen käsite sisällöltään sama kuin Copyright Actissa määritelty. Esimerkiksi GNU GPL -lisens-

<sup>246</sup> Determann 2006, s. 1431

<sup>247</sup> Carver 2005, s. 458

<sup>248</sup> Determann 2006, s. 1497



sisä paikoin viitataan Copyright Actin määritelmään ja paikoin esitetään jopa hieman ristiriitaisia selityksiä johdetun teoksen käsitteestä<sup>249</sup>. Vaikka jotkut lisenssin sanavalinnat ja FSF:n usein kysytyissä kysymyksissä annetut tulkintakannanotot näyttäisivät viittaavan Copyright Actin määritelmää laajempaan johdetun teoksen käsitteeseen, katsoo Determann GPL:n kuitenkin todennäköisesti viittaavan laissa ilmaistuun johdetun teoksen määritelmään ja ristiriitojen mahdollisesti liittyvän Copyright Actin tulkintaa koskeviin näkemyseroihin.<sup>250</sup>

Eurooppalaiset tekijänoikeuslait eivät tunne tismalleen Copyright Actissa ilmaistun kaltaista johdettua teosta. Välimäen mukaan Euroopassa johdetun teoksen luominen vastaisi lähinnä muuntelua. Välimäki katsoo eurooppalaisen vastaavan määritelmän löytyvän ohjelmistodirektiivin 4 artiklan kohdasta b, jossa mainitaan ”tietokoneohjelman kääntäminen, muuntaminen, sovittaminen ja muunlainen muuttaminen”. Tällöin kaikenlaisen tietokoneohjelman muuttamisen kattava eurooppalainen käsitys vaikuttaisi Copyright Actin määritelmää laajemmalla. Vaikka avoimen lähdekoodin lisenssit pääsääntöisesti pohjautuvat yhdysvaltalaiseen oikeuteen, voidaan lisenssien väittää kattavan myös eurooppalaisen käsityksen johdetusta teoksesta.<sup>251</sup> Koska johdetulla teoksella ei ole mitään kansainvälisesti tunnettua merkityssisältöä, voisi riskittömin vaihtoehto olla johdetun teoksen mahdollisimman yksiselitteinen määrittelemine lisenssitekstissä<sup>252</sup>.

Vastavuoroisuutta vaativilla lisensseillä voidaan lisätä ohjelmistojen avoimuutta ja mahdollistaa avoimen lähdekoodin projektien laajeneminen, kun lähdekoodia uudelleen käyttävä taho ei voi muuttaa koodia suljetuksi ja pitää tekemiään parannuksia itsellään<sup>253</sup>. Vastavuoroisuusvelvoitteet voivat kuitenkin näyttäytyä avoimen lähdekoodin komponentteja käyttävälle kehittäjälle riskinä, jos hän ei halua julkistaa itse luomaansa lähdekoodia tai levittää kokonaisuutta saman lisenssin alla kuin käyttämäänsä komponenttia. Seuraavassa alaluvussa käsitellään muun ohessa toimintatapoja, joilla vastavuoroisuutta vaativien lisenssien alaisia komponentteja on pyritty yhdistelemään omisteisiin ohjelmiin ilman, että lopputulos olisi komponentin lisenssin piiriin tuleva johdettu teos.

249 Esimerkiksi GPL 2.0 -lisenssin kohdassa 0 viitataan suoraan tekijänoikeuslaissa määriteltyyn johdettuun teokseen, kun taas kohdassa 2 viitataan tätä laajempaan jälkiperäisten teosten joukkoon (”derivative or collective works based on the Program”). ”GNU General Public License version 2”. Open Source Initiative. <https://opensource.org/licenses/GPL-2.0>. 24.11.2017.

250 Determann 2006, s. 1487–1489

251 Välimäki 2005, s. 124–125 ja Välimäki 2009, s. 211

252 Rosen 2004, s. 58

253 Dixon 2004, s. 22–23 ja Rosen 2004, s. 103–104

#### 5.4.4 Lisenssien yhteensopivuus

Kun avoimen lähdekoodin komponentteja käytetään osana laajempaa ohjelmistoa, on ratkaistava, miten eri lisenssien alaisia komponentteja voidaan keskenään yhdistellä. Lisenssien erilaiset ehdot voivat johtaa siihen, ettei komponentteja ehkä voida liittää toisiinsa lisenssiehtoja rikkomatta. Myös lisenssiehtojen tulkintaan liittyvät epäselvyydet voivat vaikeuttaa sen ratkaisemista, millaiset yhdistelmät ovat sallittuja ja millaiset eivät. Entistä haastavammaksi tilanne voi muuttua, jos komponenteista koottu kokonaisuus halutaan vielä lisensoida uudelleenlaisin ehdoin, sillä lisenssin muuttaminen on sallittua vain tekijänoikeuden haltijan luvalla – ellei lisenssi siihen erikseen oikeuta<sup>254</sup>. Vaikka ohjelmien kehittäjät pääosin ymmärtävät avoimen lähdekoodin lisenssejä melko hyvin, tulkintaongelmia aiheuttavat erityisesti tilanteet, joissa yhdistellään erilaisia lisenssejä<sup>255</sup>.

Mahdollisuuksien mukaan lisenssien yhteensopivuusongelmaa voidaan pyrkiä välttämään valitsemalla vain komponentteja, jotka on lisensoitu niin vapain ehdoin, ettei eri komponenttien yhdistely tuota vaikeuksia. Tyypillisesti edellä kuvatut sallivat lisenssit eivät juuri rajoita komponenttien käyttöä eri lisenssien alaisten ohjelmistotuotteiden kanssa, vaan esimerkiksi MIT-lisenssin alaista koodia voidaan yhdistää liki kaikkien lisenssien kanssa ja levittää kokonaisuutta uuden lisenssin alaisuudessa. Sen sijaan GPL ja muut tarttuvat lisenssit asettavat komponenttien yhdistelylle enemmän rajoitteita. Jos saatavilla on kaupallisia komponentteja, joiden tekijänoikeuden haltija on laajan OS-yhteisön sijasta jokin yksilöitävissä oleva taho, voi myös lisenssiehdoista ja niiden tulkinnasta neuvottelemisen tulla kyseeseen.<sup>256</sup> Jos taas avoimen lähdekoodin komponentti on monilisensoitu eli se on saatavilla erilaisten vaihtoehtoisten lisenssien alaisena, voi ohjelmistokehittäjä valita tarkoitukseensa parhaiten sopivan lisenssin<sup>257</sup>.

Lisenssien yhteensopimattomuutta pyritään usein ratkomaan suunniteltaessa ohjelmiston arkkitehtuuria<sup>258</sup>. Tarkoituksena on tällöin säilyttää komponenttien itsenäisyys ja välttää johdetun teoksen luomista, sillä yhteensopivuusongelmat liittyvät usein vastavuoroisuutta vaativiin lisensseihin, joiden vastavuoroisuusehto kiinnittyy edellisessä alaluvussa kuvatulla tavalla tyypillisesti johdettujen teosten julkaisemiseen ja levittämiseen.

Ilmeisintä lisenssien yhteensopimattomuus on tilanteessa, jossa hankittua avointa lähdekoodia sijoitetaan suoraan yhteensopimattoman lisenssin alla levitettävän koodin jouk-

254 Wu ym. 2017, s. 1195

255 Almeida – Murphy – Wilson – Hoyer 2017, s. 5

256 Hammouda – Mikkonen – Oksanen – Jaaksi 2010, s. 209 ja Välimäki 2009, s. 224

257 German – Hassan 2009, s. 195

258 Hammouda ym. 2010, s. 209

koon<sup>259</sup>. Tilanne vaikuttaisi melko selkeästi sopivan sekä yhdysvaltalaiseen (17 U.S.C. § 101) että eurooppalaiseen (ohjelmistodirektiivin 4 art. kohta b) käsitykseen johdetusta teoksesta, koska yhdeksi kokonaisuudeksi yhdistellyt lähdekoodin osat eivät tällöin ole itsenäisiä suhteessa toisiinsa. Miten tahansa lyhyen koodinpätkän kopiointi uuteen yhteyteen ei kuitenkaan riittäne. Yhdysvaltalaisen käsityksen mukaan johdetun teoksen tulee pohjautua (”based upon”) johonkin olemassa olevaan teokseen eli johdetun teoksen tulee sisältää merkittäviä ja luovia osuuksia aiemmasta teoksesta<sup>260</sup>. Eurooppalainen määritelmä ei sinänsä aseta vastaavaa vaatimusta, vaan lyhyemmänkin koodin sijoittaminen toisen koodin sekaan voisi merkitä johdetun teoksen luomista<sup>261</sup>. Kuitenkaan miten tahansa lyhyet ja yleisesti käytetyt ilmaukset eivät saa tekijänoikeussuojaa, joten jonkinlaista itsenäisyyttä ja omaperäisyyttä omaksuttavalta lähdekoodilta edellytetään, jotta kappaleiden valmistamista voitaisiin pitää tekijän yksinoikeutena. Riittävän suurissa määrin toteutettu leikkaa-liitä-tyyppinen lähdekoodien sekoittaminen ohjelmointityössä voi siis johtaa lisenssien ristiriitaiseen yhdistelyyn tai velvoittaa myös itse tuotetun lähdekoodin lisensoimiseen vastavuoroisuusehdon edellyttämällä tavalla.

Avoimen lähdekoodin hyödyntäminen osana ohjelmistokehitystä ei kuitenkaan välttämättä merkitse eri lähteistä hankittujen lähdekoodien sekoittumista, vaan avoimen lähdekoodin komponentteja voidaan myös hyödyntää muokkaamattomina ja muuhun kokonaisuuteen nähden ulkoisina<sup>262</sup>. Komponentti säilyy itsenäisenä, mutta ohjelmisto käyttää jotain komponentin toiminnallisuutta. Tällainen ratkaisu ei muodosta johdettua teosta, koska tekijänoikeus ei kata käyttöä tai rajapintoja.<sup>263</sup> Näin ollen myös vastavuoroisuutta vaativien lisenssien alaisia ohjelmia voidaan tietysin edellytyksin levittää jopa yhdessä omisteisten ohjelmien kanssa. GPL-lisensioitujen ohjelmien suhteen FSF muotoilee asian niin, että GPL:n alaisten ohjelmien ja omisteisten ohjelmien täytyy kommunikoida ”käsi-varren mitan” päässä toisistaan, eivätkä ne saa muodostaa yhdessä uutta ohjelmaa<sup>264</sup>.

Tiettyjen mekanismien suhteen vallitsee kuitenkin eriäviä tulkintoja siitä, onko ohjelmien välinen riippuvuussuhde niin vahva, että kyse on johdetun teoksen luomisesta. Erimielisyyttä on herättänyt erityisesti GPL:n kohdalla linkittäminen. Rosenin mukaan johdetun teoksen luominen edellyttää aina lisensoidun teoksen muuntelemista, minkä takia läh-

259 Ruffin – Ebert 2004, s. 84

260 Determann 2006, s. 1497

261 Välimäki 2005, s. 125

262 Erilaisista komponenttien integrointikeinoista ks. esim. Jansen – Brinkkemper – Hunink – Demir 2008, s. 42

263 Välimäki 2005, s. 128

264 ”Frequently Asked Questions about the GNU Licenses.” Free Software Foundation. <https://www.gnu.org/licenses/gpl-faq.html>

dekoodin muuntelemista edellyttävä staattinen linkittäminen tarkoittaa johdetun teoksen luomista. Sen sijaan dynaaminen linkitys ei vaadi ohjelman lähdekoodiin kajoamista, jolloin Rosenin mukaan ei myöskään ole kyse johdetun teoksen luomisesta.<sup>265</sup> Sen sijaan FSF itse katsoo, että niin staattisen kuin dynaamisenkin linkittämisen tapauksessa GPL laajenee kattamaan koko yhdistelmän<sup>266</sup>. Eräänlaisena vastauksena linkittämistä koskevaan ongelmaan voidaan pitää LGPL-lisenssiä, joka nimenomaisesti sallii linkittämisen ilman, että kokonaisuus pitäisi saattaa saman lisenssin alaisuuteen.

Kysymys siitä, milloin yhdessä toimivat ohjelmat säilyvät erillisinä ja milloin ne muodostavat johdettuna teoksena pidettävän kokonaisuuden, onkin jossain määrin säilynyt ratkaisemattomana. Komponenttien lisenssien yhteensopimattomuuden ratkaisemiseksi käytettyjä arkkitehtonisia keinoja tutkineet Hammouda ym. painottavatkin, ettei täyttä varmuutta keinojen juridisesta kestävydestä ole, vaikka keinot ovat yleisesti käytettyjä<sup>267</sup>. FSF toteaa kommunikoivien ohjelmien suhteen luonteen jäävän viime kädessä tuomioistuimen ratkaistavaksi, joskin FSF pitää ratkaisevina kriteereinä ohjelmien keskinäistä kommunikaatiomekanismia ja kommunikaation semantiikkaa (millaista informaatiota ohjelmat vaihtavat)<sup>268</sup>.

Vaikka komponenttien lisenssien yhteensopimattomuutta ei saataisi ratkaistua, voi komponentteja olla mahdollista yhdistellä vapaasti, kunhan kokonaisuutta ei levitetä. Myös GPL sallii ohjelmien yhdistelemisen ja johdannaisten teosten luomisen, kunhan kyse on loppukäyttäjän yksityisestä tai yrityksen sisäisestä käytöstä.<sup>269</sup> FSF viittaa levittämisen käsitteen olevan riippuvainen sovellettavasta tekijänoikeuslaista<sup>270</sup>. Edellä alaluvussa 4.4.1 todetulla tavalla voisi olla teoriassa mahdollista toteuttaa kokeilu niin, ettei kyse olisi tekijänoikeuslain tarkoittamasta yleisön saataviin saattamisesta. Jos toiminta ei kohdistuisi yleisöön, olisi avoimen lähdekoodin huolettomampi yhdistely ehkä mahdollista. Jos kyse kuitenkin on jo edistyneemmästä prototyypistä, jota kehitetään kohti yleisön saataviin saatettavaa tuotetta, asian käytännön merkitys lienee pieni. Jos yleisön saataviin saatettava tuote on tarkoitus luoda prototyyppiä edelleen hiomalla, lienee komponenttien lisenssiehtojen mukainen yhdistely järkevää jo varhaisemmassa vaiheessa, jotta muutoksia ei jouduta tekemään myöhemmin.

265 Rosen 2001, s. 2. Ks. myös Determann 2006, s. 1497, joka ei myöskään pidä dynaamista linkittämistä johdetun teoksen luomisena. Hammouda ym. 2010, s. 211, mainitsevat dynaamisen linkittämisen keinona lisenssien yhteensopimattomuuden välttämiseen.

266 "Frequently Asked Questions about the GNU Licenses." Free Software Foundation. <https://www.gnu.org/licenses/gpl-faq.html>

267 Hammouda ym. 2010, s. 214

268 "Frequently Asked Questions about the GNU Licenses." Free Software Foundation. <https://www.gnu.org/licenses/gpl-faq.html>

269 Determann 2006, s. 1491 ja German – Hassan 2009, s. 192

270 "Frequently Asked Questions about the GNU Licenses." Free Software Foundation. <https://www.gnu.org/licenses/gpl-faq.html>

#### 5.4.5 Lisenssien vastuunrajoituslausekkeet

Avoimen lähdekoodin ohjelmat sisältävät usein eri lähteistä kopioitua koodia, jota uudelleen käyttämällä on saatu aikaan uusi ohjelma<sup>271</sup>. Tutkimuksissa koodin on katsottu varsin yleisesti kulkeutuvan yhteisöllisten ohjelmointialustojen projekteista toisiin, joskin ohjelmoijat vaikuttaisivat pääsääntöisesti osaavan toimia lisenssien mukaisesti toisen lähdekoodia käyttäessään<sup>272</sup>. Kuitenkin toisen tekijänoikeuden alaista materiaalia hyödynnettäessä voidaan jopa täysin tiedostamatta syyllistyä tekijänoikeuden loukkaukseen, jos lähdekoodia kopioidaan tekijänoikeuslain vastaisesti tai jos lisenssiehtoja ei noudateta<sup>273</sup>. Vaikka avointa lähdekoodia käyttävä ohjelmistokehittäjä haluaisikin olla varovainen käyttämiensä komponenttien suhteen, saattaa lähdekoodin osien alkuperän selvittäminen olla lähes mahdotonta<sup>274</sup>.

Tuotteita, materiaalia tai oikeuksia hankittaessa suositellaan aina vaadittavan luovuttajalta sopimukseen vakuutus siitä, että luovuttaja on luovutettavien oikeuksien haltija. Jos luovuttaja sopimuksessa antaa tällaisen vakuutuksen, on hän vastuussa kolmannen oikeuden loukkauksesta aiheutuvista vahingoista sopimuskumppanille.<sup>275</sup> Avoimen lähdekoodin lisenssien kohdalla lisenssinantaja ei kuitenkaan tyypillisesti tällaista vakuutusta anna, vaan päinvastoin lisenssit usein sisältävät ehdon, jolla riskit siirretään lisenssinsaajalle<sup>276</sup>. Vastuunrajoituslausekkeella lisenssinantaja voi sulkea pois vastuunsa lisenssinsaajaa kohtaan, mutta se ei vaikuta lisenssinantajan ja loukatun oikeuden haltijan suhteeseen<sup>277</sup>.

Avoimen lähdekoodin ohjelmiin oikeudet annetaan yleensä ”as is” -tyyppisellä lisenssillä, jossa lisenssinantaja vapauttaa itsensä niin teknisiin kuin oikeudellisiin virheisiin liittyvistä vastuista<sup>278</sup>. Lisenssinantaja ei siis vastaa ohjelmaan sisältyvän teknisen virheen aiheuttamasta vahingosta, muttei myöskään kolmannen osapuolen immateriaalioikeuksien loukkauksesta aiheutuvista vahingoista. Jos lisenssinsaaja kärsii vahinkoa em. virheiden johdosta, ei hänellä ole mahdollisuutta periä lisenssinantajalta korvausta<sup>279</sup>.

Jos ohjelmaan on päässyt livahtamaan luvatta kopioitua koodia, ei vastuussa ole vain kolmannen oikeutta loukkaavan koodinpätkän ohjelman lähdekoodiin ensi kertaa kopioinut henkilö, vaan kaikki kyseistä ohjelmaa käyttävät, jakelevat ja muokkaavat tahot voivat

271 Ruffin – Ebert 2004, s. 84

272 German ym. 2009, s. 88 ja Gharehyazie ym. 2017, s. 300

273 Välimäki – Laine 2004, s. 902 ja Välimäki 2005, s. 167

274 Ruffin – Ebert 2004, s. 84

275 Ashorn 2003, s. 86

276 Dixon 2004, s. 87

277 Välimäki 2005, s. 173

278 Välimäki – Laine 2004, s. 901

279 Ruffin – Ebert 2004, s. 84

syyllistyä tekijänoikeuden loukkaukseen<sup>280</sup>. Tekijänoikeuden loukkaukseen liittyvä hyvitysvastuu ei TekijäL 57.1 §:n mukaan edellytä tahallisuutta tai edes tuottamusta, joten vilpittömässäkin mielessä ollut toimija voi joutua suorittamaan tekijänoikeuden haltijalle lain edellyttämän kohtuullisen hyvityksen. Myöskään ohjelmistopatenttiin kohdistuva loukkaus ei edellytä tahallisuutta tai tuottamusta, vaan patenttilain (PatL, 15.12.1967/550) 58.2 §:n mukaan vilpittömässäkin mielessä ollut loukkaaja voi joutua suorittamaan korvusta siinä määrin kuin kohtuulliseksi katsotaan. Lisenssinsaaja ei siis vältty vastuulta, vaikkei hän olisi ollut tietoinen avoimen lähdekoodin komponentin sisältämästä kolmannen immateriaalioikeutta loukkaavasta materiaalista.

Koska laki käytännössä asettaa immateriaalioikeuksien käyttäjälle selonottovelvollisuuden toimintansa luvallisuudesta (eikä avoimen lähdekoodin lisensseissä tyypillisesti anneta mitään vakuutuksia kolmansien oikeuksien suhteen), voisi lisenssinsaajan kannalta olla tärkeää pystyä jollain lailla varmistumaan lähdekoodin alkuperästä. Lähdekoodin tutkiminen on kuitenkin vaivalloista ja harvoin täydellistä varmuutta sen lähteiden luvallisuudesta voidaan saavuttaa. Huolellisesti toimiva ohjelmistokehittäjä voi pyrkiä suorittamaan vertailua käyttämänsä komponentin ja kopioinnin kohteeksi epäilemänsä lähdekoodin välillä, jos tällainen lähdekoodi on saatavilla. Myös kokonaisuuden sisältämien eri komponenttien lisenssien yhteensopivuutta voidaan tutkia, jos lisenssit ovat tiedossa. Jos lähdekoodiin tai lisenssiehtoihin ei ole pääsyä, ei kehittäjä voi juuri mitenkään pyrkiä selvittämään komponentteihin liittyviä oikeudellisia riskejä.<sup>281</sup>

Manner pitääkin kolmansien immateriaalioikeuksien loukkausta merkittävänä avoimen lähdekoodin käyttöön liittyvänä riskinä ja näkee ongelman entisestään kärjistyvän ohjelmistopatenttien yleistytessä<sup>282</sup>. Avoimen lähdekoodin kohdalla myös juuri lähdekoodin avoimuus lisää riskiä jäädä kiinni kolmannen osapuolen koodin luvattomasta hyödyntämisestä, koska kolmansilla on mahdollisuus tarkastella ohjelman lähdekoodia ja havaita koodin kopioiminen. Kolmannen oikeuden loukkaus voi aiheuttaa vilpittömässäkin mielessä toimivalle lisenssinsaajalle suurta taloudellista vahinkoa, ellei lisenssinsaaja ole osannut varautua mahdollisiin loukkauksiin vakuutuksella.<sup>283</sup>

---

280 Manner 2001

281 Välimäki – Laine 2004, s. 902–903

282 Manner 2001

283 Välimäki – Laine 2004, s. 902–903 ja Välimäki 2009, s. 168

## 5.5 Strategisia näkökohtia

Uutta tuotetta kehittäessään yrityksen on valittava, kehittääkö se immateriaalioikeutensa itse vai hankkiiko se oikeudet ulkopuolisilta. Jos joku toinen on jo kehittänyt halutun tuotteen tai ominaisuuden aiemmin, ei resurssien suuntaaminen omaan vastaavaan kehitystyöhön välttämättä ole mielekäästä, vaan oikeus toisen immateriaalioikeuden hyödyntämiseen voidaan ehkä hankkia lisensioimalla. Lisensiointiin voi liittyä merkittäviäkin transaktiokustannuksia, mutta avoimen lähdekoodin lisenssien ja Creative Commons-lisenssien kaltaisia julkisia lisenssejä käytettäessä lisensioinnin aiheuttamat kustannukset pysyvät alhaisina. Vaikka valmiin materiaalin hyödyntäminen voi olla halvempaa ja nopeampaa, kannattaa vaihtoehtoja kuitenkin punnita juuri kyseistä tuotekehitystilannetta silmällä pitäen. Jos itse toteuttamalla tuotteeseen voidaan luoda erottuvia ja omaperäisiä ominaisuuksia, kannattaisiko ainakin ko. ominaisuudet kehittää yrityksen omin voimin?

Prototyyppiä varten ei välttämättä kannata hankkia tai kehittää enempää materiaalia kuin kokeilussa testattavien hypoteesien kannalta on välttämätöntä. IPR-strategiaan kuuluu myös oikeuksien oikeasuhtainen hankinta, sillä ei ole järkevää hankkia sellaisia oikeuksia, joita ei tarvita. Prototyypointi on nopeampaa ja edullisempaa, jos jokaista valmiiseen tuotteeseen tulevaa ominaisuutta ei pyritä toteuttamaan täydellisesti jo prototyyppiin. Ominaisuuksien puutteellisuus mahdollistaa myös kokeilun fokuksen suuntaamisen tiettyihin rajattuihin osa-alueisiin.

Ohjelmistokehityksen alkuvaiheessa prototyyppiä varten ei ole välttämätöntä kirjoittaa riviäkään koodia, vaan ensimmäiset kokeilusykliä voidaan viedä läpi esimerkiksi paperiprototyypin tai mockupin avulla. Näiden prototyyppien muokkaaminen ei vaadi lukuisia työtunteja tai teknistä ammattitaitoa. Tällöin voidaan myös vielä välttyä mahdollisilta lisensiointikuluilta ja avoimen lähdekoodin käyttämiseen liittyviltä riskeiltä. Edullisuutensa ja riskittömyytensä vuoksi tällaiset prototyypointimenetelmät sopivat ensimmäisiin kokeiluihin hyvin, mutta kovin edistyneiden toimintojen havainnollistaminen voi olla vaikeaa. Loppuvaiheen kokeiluissa luultavasti jo tarvitaan erityyppistä prototyyppiä oikeanlaisen palautteen saamiseksi.

Luvussa on tehty selkoa erityisesti avoimen lähdekoodin hyödyntämisen mahdollisuuksista ja riskeistä, sillä avointa lähdekoodia käytetään nykyään varsin yleisesti nopeutta vaativassa ohjelmistokehityksessä. Koska valmiit komponentit voivat selkeästi lyhentää tuotekehitysprosessia ja avoin lähdekoodi on usein jopa ilmaista, voisi avoimen lähdekoodin helposti nähdä myös osana ohjelmistoalan yrityksen toteuttamaa kokeilua. Avoimuuden laajempikin



omaksuminen osaksi strategiaa voisi sopia kokeiluja hyödyntävälle yritykselle (ks. 7.3).

Avoimen lähdekoodin komponenttien käyttämiseen liittyy kuitenkin myös riskejä, jotka realisoituessaan voivat aiheuttaa yritykselle merkittävää taloudellista vahinkoa. Jos yritys päättää rakentaa tuotteensa avoimen lähdekoodin komponenteista, jotka nekin on luotu komponentteja yhdistellen, voi kaikkien eri lisenssien yhteensopivuuden hahmottaminen ja koodin alkuperän varmistaminen olla hyvin työläs tehtävä. Yleensä avoimen lähdekoodin kohdalla lisenssinantaja ei anna takuita ohjelman teknisestä toimivuudesta tai kolmannen oikeuksien loukkaamattomuudesta, joten huolellisen ohjelmistokehittäjän kannattaa tutustua ohjelmaan ja sen lisenssiin hyvin ennen avoimen lähdekoodin levittämistä osana omaa ohjelmistoaan.

Kolmannen oikeuksiin kohdistuvien loukkausten lisäksi avoimen lähdekoodin käyttö voi loukata myös lisenssinantajan oikeutta, jos lisenssiehtoja ei noudateta. Sallivilla lisensseillä lisenssinantaja antaa lisenssinsaaajalle yleensä varsin laajat oikeudet, mutta erityisesti vastavuoroisuutta vaativat lisenssit sisältävät ehtoja, jotka ovat monien kaupallisten ohjelmistokehittäjien näkökulmasta hyvin rajoittavia. Jos vastavuoroisuutta vaativan lisenssin alaisesta ohjelmasta luodaan johdettu teos, voi yritys joutua vasten tahtoaan julkaisemaan omaa arvokasta lähdekoodiaan tai ottamaan riskin lisenssiehtojen rikkomisesta. Ohjelmistoarkkitehtuurin huolellisella suunnittelulla johdetun teoksen luominen vastavuoroisuutta vaativan lisenssin alaisesta komponentista voidaan välttää, joskin lisenssien sanavalintoihin liittyvien tulkintaerojen vuoksi joidenkin menetelmien juridinen kestävyys on kyseenalainen. Avoimen lähdekoodin liittäminen jonkin patentoidun keksinnön toteuttamaan ohjelmaan voi myös olla ongelmallista, sillä monet avoimen lähdekoodin lisenssit vaativat tällaisten keksintöjen lisensoimista ilmaiseksi<sup>284</sup>.

Avoimen lähdekoodin hyödyntäminen voi tuoda monia etuja, mutta sen käyttäminen vaatii suunnitelmallisuutta. Useissa yrityksissä onkin asetettu sisäisiä sääntöjä avoimen lähdekoodin käytölle<sup>285</sup>. Suositeltavaa on pitää kirjaa kaikesta käytettävästä koodista, jotta avoin lähdekoodi ja siihen liittyvät lisenssit voidaan jäljittää. Vaikka avointa lähdekoodia käytettäisiin sinänsä lisenssiehtojen mukaisesti, eivät kaikki käyttötavat ole OS-yhteisön silmissä eettisesti suositeltavia. Bayersdorfer kehottaa avointa lähdekoodia hyödyntäviä yrityksiä ”olemaan kilttejä yhteisölle”, mikä voi merkitä esimerkiksi johdettujen teosten tai muun koodin julkaisemista yhteisössä tai avoimen lähdekoodin projektin taloudellista tukemista.<sup>286</sup>

284 Välimäki 2009, s. 223

285 Välimäki 2009, s. 221

286 Bayersdorfer 2007, s. 33–34



## 6. TEKIJYYS JA TEKIJÄNOIKEUDEN SIIRTYMINEN KOKEILUSSA

### 6.1 Tekijänoikeuden subjekti ja tekijänoikeudellinen yhteistyö

Tekijäl 1 § osoittaa tekijänoikeuden alkuperäiseksi haltijaksi eli tekijäksi sen, ”joka on luonut kirjallisen tai taiteellisen teoksen”. Tällöin tekijyyden määrittelyn kannalta olennaiseksi nousee se, voidaanko tekijän suorittamat toimet katsoa luomiseksi. Luominen on toimintaa, jonka seurauksena syntyy teoksen määritelmän täyttävä henkisen työn tuote. Näin ollen esimerkiksi jonkin objektin löytäminen ei ole luomista.<sup>287</sup> Suomalaisessa tekijänoikeudessa yksinoikeus teokseen syntyy aina luonnolliselle henkilölle, joten tekijä voi olla vain ihminen – ei esimerkiksi tietokone tai oikeushenkilö<sup>288</sup>. Tämä on vallitseva näkemys useimmissa muissakin romaanis-germaaniseen oikeusryhmään kuuluvissa maissa, kun taas common law -maissa tekijyys ei ole yhtä tiukasti rajattu vain luonnollisiin henkilöihin<sup>289</sup>. Suomalaisessa tekijänoikeudessa oikeushenkilökin voi tulla tekijänoikeuden haltijaksi, mutta sen oikeus on aina derivatiivinen eli luonnolliselta henkilöltä johdettu<sup>290</sup>. Tekijänoikeus tulee siis tällöin siirtää luonnolliselta henkilöltä oikeushenkilölle.

Tekijänoikeus voi myös syntyä yhteisesti useammalle kuin yhdelle tekijälle. Tekijäl 6 §:ssä säädetään yhteisteoksesta, joka on kyseessä silloin, kun vähintään kaksi tekijää on yhdessä luonut teoksen siten, etteivät heidän osuutensa muodosta itsenäisiä teoksia. Kunkin tekijän panoksen on oltava luova, mutta panokset voivat olla määrällisesti ja laadullisesti eritasoisia<sup>291</sup>. Yhteistyö voi olla joko samaan aikaan tapahtuvaa (horisontaalinen yhteistyö) tai peräkkäistä (vertikaalinen yhteistyö). Yhteistyö on vertikaalista silloin, kun toinen tekijä jatkaa tai täydentää toisen aloittamaa työtä.<sup>292</sup> Ryhmässä tapahtuvassa ohjelmoinnissa yhteisteos lienee tavallisin tekijänoikeudellisen yhteistyön muoto, sillä ohjelmien kohdalla eri tekijöiden luomistyön tuloksia on usein vaikea eritellä valmiista tuotteesta<sup>293</sup>.

Yhteisteoksen tekijöillä on yhteisesti tekijänoikeus luomaansa teokseen, mikä tarkoittaa sitä, että myös lupa teoksen käyttämiseen tarvitaan jokaiselta tekijältä – elleivät tekijät

287 Haarmann 1992, s. 80

288 Haarmann – Mansala 2012, s. 91, Harenko ym. 2016, s. 15–16 ja KM 1987:8, s. 65. Komiteamietinnössä 1987:8, s. 65, tietokoneohjelma on katsottu tekniseksi apuvälineeksi, jota tekijä käyttää luomistyössään.

289 Goldstein – Hugenholtz 2013, s. 248

290 Haarmann 1992, s. 80

291 Goldstein – Hugenholtz 2013, s. 250–251

292 Kivimäki 1966, s. 52 ja Koivumaa 1995, s. 131

293 Koivumaa 1995, s. 31 ja Välimäki 2009, s. 28

ole keskenään muuta sopineet<sup>294</sup>. Tapauksessa, jossa yhteisteoksen tekijät eivät ole sopimuksin järjestäneet välejä, tulevat sovellettavaksi yleiset siviilioikeudelliset, erityisesti yhteisomistusoikeutta koskevat periaatteet<sup>295</sup>. TekijäL 6 §:n mukaan kullakin tekijällä on kuitenkin itsenäinen oikeus vaatimusten esittämiseen oikeuden loukkaamisen johdosta.

Jos teoksen tekijöiden panokset ovat toisistaan eroteltavissa olevia itsenäisiä teoksia, on kyse yhteisteoksen sijaan yhteenliitetystä teoksesta<sup>296</sup>. Tällaisessa tapauksessa kukin tekijä on muista riippumattomasti toteuttanut osansa ja määrää omasta teoksestaan itse, vaikka teokset näyttäytyvätkin kokonaisuutena.<sup>297</sup> Lausunnossa TN 1992:3 videopeli katsottiin yhteenliitettyksi teokseksi, jonka sisältämien tietokoneohjelman, musiikin ja kuvien tekijöille kuului kullekin tekijänoikeus oman panoksensa osalta. Yhteenliitetty teos voisi olla myös esimerkiksi useita tietokoneohjelmia sisältävä ohjelmapaketti<sup>298</sup>.

Ohjelmistoteollisuuden kannalta keskeisiä tekijänoikeudellisen yhteistyön muotoja ovat myös TekijäL 4 §:n mukainen muunnelma sekä 5 §:n tarkoittama kokoomateos<sup>299</sup>. TekijäL 4–5 §:ää on jo tarkemmin käsitelty kohdassa 4.4.2.

Kuten ohjelmistotuotannossa ylipäänsä, edellä mainitut tekijänoikeudellisen yhteistyön muodot voivat tulla kyseeseen myös ohjelmistoa kokeiluun kehitettäessä. Koska yhteistyönä luotujen teosten hyödyntäminen kokonaisuutena edellyttää kaikkien tekijöiden suostumusta, on tekijyyssuhteiden hahmottaminen ja oikeuksien siirtymisen varmistaminen teosta hyödyntämään pyrkivälle taholle keskeistä. Seuraavassa alaluvussa tarkastelen kokeiluun osallistuvien luonnollisten henkilöiden välisiä suhteita ja tekijyyden määrittämistä koehenkilöiden ja kehitystiimin yhteistyössä (6.2). Alaluvussa 6.3 käsittelen tekijänoikeuden siirtymistä kokeiluun osallistuvien luonnollisten henkilöiden ja kokeilu- ja tuotekehityksessään hyödyntävän oikeushenkilön välillä. Lopuksi esitän yhteenvedon toimintatavoista, joilla ohjelmistoalan yritys voi luoda ennakoitavuutta tekijänoikeuksien syntymiseen ja siirtymiseen kokeiluja käyttäessään (6.4).

---

294 Harenko ym. 2016, s. 85

295 KM 1953:5, s. 51

296 Esimerkiksi tapauksessa KKO 1956 II 76 operetin sävellys ja teksti katsottiin erillisiksi teoksiksi, joiden tekijöillä oli kummallakin tekijänoikeus omaan teokseensa. Kivimäki 1966, s. 54, mainitsee muina tyypillisinä yhteenliitettyinä teoksina oopperat, oratoriot ja kuvitetut kirjat.

297 Haarmann 1992, s. 84 ja Kivimäki 1966, s. 49

298 Koivumaa 1995, s. 131

299 Välimäki 2009, s. 27

## 6.2 Tekijyyden määrittäminen kokeilussa

### 6.2.1 Koehenkilöiden ja kehitystiimin rooleista

Innovaatiot ovat harvoin yhden ihmisen keksintöjä. Uusien mullistavien tuotteiden luominen vaatii tyypillisesti eri henkilöiden välistä yhteistyötä, jossa ideoita jaetaan ja yhdistellään.<sup>300</sup> Tällaiseen ideoiden jakamiseen ja yhdistelyyn perustuu myös kokeilu. Kokeilu poikkeaa perinteisestä tuotekehitysprosessista kuitenkin muun muassa siten, että kokeilussa prosessiin otetaan mukaan joko aitoja asiakkaita tai koehenkilöitä luvussa 2 esitetyllä tavalla. Kehitteillä olevasta ohjelmistosta riippuen koehenkilöt voivat olla esimerkiksi kulluttajia tai ohjelmistoyrityksen asiakasyrityksen edustajia. Pilotoinnista poiketen kokeilu ei ole valmiin tuotteen testausta, vaan koehenkilöt tulevat mukaan jo varhaisemmassa vaiheessa tuotekehitystä<sup>301</sup>.

Koehenkilöiden osallistaminen voi antaa ohjelmistoyritykselle tietoa todellisista käyttäjätarpeista ja käyttäjien mieltymyksistä, mutta toisaalta koehenkilön innovointiin antama panos voi monimutkaistaa kehitteillä olevan ohjelman tekijänoikeudellisia suhteita. Ketterien toimintatapojen riskinä voidaankin pitää epävarmuutta siitä, kenelle tekijänoikeus ohjelmaan tai sen osaan syntyy: ohjelmistoyrityksen työntekijälle, ohjelman toteuttamiseen osallistuvalla asiakkaalla vai kenties ohjelman testaajalle<sup>302</sup>. Pelkkä osallistuminen innovaatioprosessiin ei kuitenkaan vielä riitä tekijän aseman saavuttamiseen, vaan tekijäksi katsottavan henkilön panoksen prosessissa tulee olla luova ja teoksen syntymiseen johtava. Tällaisiksi luoviksi toimijoiksi ei katsota esimerkiksi teknisiä avustajia, jotka eivät näin ollen saa tekijänoikeutta luotuun kohteeseen<sup>303</sup>. Kokeiluun osallistuva koehenkilö ei kuitenkaan rinnastu roolinsa puolesta tekniseen avustajaan, sillä hänen tehtävänsä ei ole auttaa teoksen teknisessä toteuttamisessa tekijän sanelemalla tavalla.

Hassi ym. näkevät koehenkilön mm. ”ideoiden ja vinkkien” antajana sekä ”kontaktien, osaamisen ja resurssien tarjoajana”<sup>304</sup>. Ensimmäinen luonnehdinta viittaa koehenkilöön lähinnä ideatasolla työskentelevänä toimijana, joka antaa sanallista palautetta kehitteillä olevasta tuotteesta. Työsuhtetekijänoikeutta käsittelevissä tekijänoikeuslain esitoissa mainitaan, että tekijän työtä ohjaavalla työnantajan edustajalla voi olla huomattava vaikutus tekijän luoman teoksen muotoon ja sisältöön, mutta välittämistään ”ideoista ja im-

300 Gollin 2008, s. 97

301 Poskela ym. 2015, s. 12

302 Sandvik 2011

303 Haarmann 2014, s. 68

304 Hassi ym. 2015, s. 11

pulsseista” huolimatta työnantajan edustajaa ei välttämättä katsota tekijäksi<sup>305</sup>. Esitöissä ilmaista voitaisiin soveltaa myös koehenkilöön luovan työn kannalta keskeisten ideoiden välittäjänä: häntäkään ei voida välttämättä pitää tekijänä, vaikka hänen osallistumisensa olisi lopputuleman kannalta kriittistä. Toisaalta Hassin ym. näkemys siitä, että koehenkilö voi tuoda kokeiluun myös omaa erityisosaamistaan, saa koehenkilön näyttäytymään teoksen luomisen suhteen aktiivisemmassa roolissa. Tätä osaamistaan koehenkilö voisi kuitenkin tuoda kokeiluun monin tavoin: ehkäpä prototyyppiä kommentoiden, ehkäpä jopa sitä muokaten.

Koehenkilöiden lisäksi kokeiluun osallistuu joukko kokeilun toteuttamisesta vastaavia henkilöitä, joita tässä tutkielmassa kutsutaan Hassin ym. (2015) omaksumalla tavalla kehitystiimiksi. Kehitystiimin kokoonpanon olisi suotavaa olla monialainen, sillä teknologisen osaamisen ohella kokeilussa eduksi on esimerkiksi asiakasrajapintaa, käyttäytymistieteitä ja markkinointia koskeva asiantuntemus<sup>306</sup>. Tässä tutkielmassa saman oikeushenkilön (ohjelmistoalan yrityksen) oletetaan sekä toteuttavan kokeilun että käyttävän kokeilun tuloksia tuotekehityksessään, vaikka todellisuudessa kokeilun toteuttaminen voidaan myös antaa tehtäväksi jollekin toiselle yritykselle, tutkimuslaitokselle tai oppilaitokselle<sup>307</sup>. Tällöin koehenkilöiden ja kehitystiimin sekä koehenkilöiden, kehitystiimin ja kokeillen tuotettaan kehittävän yrityksen välisten suhteiden lisäksi tarkasteltavaksi tulisi kokeilun toteuttavan tahon ja kokeilun tuloksia hyödyntävän tahon välinen suhde, niiden oikeudet käyttää toistensa immateriaalioikeuksia ja tietotaitoa sekä kokeilussa syntyvien immateriaalioikeuksien siirtäminen kokeilun tuloksia hyödyntävälle taholle.

Mahdollisesti myös kehitystiimin jäsenillä on kokeilussa omat yksilölliset roolinsa – erityisesti jos tiimi on koottu eri alojen asiantuntijoista. Osa kehitystiimin jäsenistä voi esimerkiksi olla vastuussa prototyypin valmistamisesta ja muokkaamisesta kokeilun kestäessä, kun taas joidenkin tehtävät voivat liittyä pikemminkin koehenkilöiden kanssa kommunikamiseen ja kokeilun tulosten analysoimiseen. Tällaisessa tapauksessa kaikista kehitystiimin jäsenistäkään ei tule kokeilussa syntyvän teoksen tekijöitä, koska kaikkien rooli ei

305 KM 1987:7, s. 125

306 Poskela ym. 2015, s. 16

307 Ks. esim. Heikkinen – Österberg 2012, s. 15, joiden mukaan living lab -ekosysteemissä on viisi erilaista toimijaa: käyttäjät, hyödyntäjät, kehittäjät, mahdollistajat ja living lab -operaattorit. Kehittäjät tarjoavat menetelmiä, työkaluja ja resursseja käyttäjiä osallistavan casen toteuttamiseksi, kun taas hyödyntäjät käyttävät casesta saatua palautetta esim. tuotteensa tai palvelunsa kehittämiseen voidaan tosin itsekkin osallistua aktiivisesti casen toteuttamiseen. Myös monissa Hassin ym. 2015 esittelemissä kokeilujen case-esimerkeissä kehittäjä ja hyödyntäjä ovat ainakin jossain määrin eroteltavissa toisistaan kehitystiimin koostuessa erilaisten organisaatioiden edustajista ja kokeilun toteuttamisen perustuessa erilaisten toimijoiden yhteistyöhön. Ks. esim. Hassi ym. 2015, s. 5–6, 13, 83–86, 114–117. Vrt. Ries 2016, jonka antamissa useissa esimerkeissä (mm. s. 62–65, 76–77, 83–87, 117–119) kehittäjä ja hyödyntäjä näyttäisivät olevan sama taho. Moni Riesin esimerkeistä koskee startup-yrityksiä, joille tämä lieneekin luonteva ratkaisu.

ole riittävän suorassa yhteydessä teoksen luomiseen. Selvää kuitenkin on, että kokeilussa voisi olla kyse jostain tekijänoikeudellisen yhteistyön muodosta, jolloin useammalle luovaan toimintaan osallistuvalla henkilöllä (mahdollisesti niin kehitystiimin jäsenelle kuin koehenkilöllekin) saattaisi muodostua tekijänoikeus kokeilussa syntyvään tuotteeseen.

### **6.2.2 Tekijänoikeudellisesti relevantti toiminta**

Takki jakaa tietokoneohjelman kehitysprosessin kolmeen vaiheeseen: ideoiden luomiseen, ohjelman rakenteen luomiseen ja ohjelman koodin luomiseen. Näistä vaiheista ainoastaan viimeisen, koodaamisen, tulokset saavat tekijänoikeudellista suojaa.<sup>308</sup> Tekijänoikeusneuvosto tekee lausunnossaan 2007:3 vastaavan jaon nimetessään ohjelmistokehityksen vaiheiksi ideatasoisen suunnittelun ja sitä seuraavan tekijänoikeudellisesti suojatun tietokoneohjelman syntymisen. Ideatason suunnitteluun sisältyvät tekijänoikeusneuvoston mukaan esitutkimus, vaatimusten määrittely ja toiminnallinen määrittely, kun taas tekijänoikeudellisesti merkityksellistä toimintaa ovat ohjelman tekninen suunnittelu ja varsinainen ohjelmointityö. Lausunnossa kuitenkin mainitaan vaiheiden välisen rajan voivan olla epäselvä.

Tekijänoikeusneuvosto on lausuntokäytännössään todennut, että tietokoneohjelman tekijänä voidaan pitää joko systeemin suunnittelijaa, ohjelmoijaa tai molempia. Systeemin suunnittelijana neuvosto pitää ohjelman toteuttamisen eri ratkaisuvaihtoehtojen harkinnan ja valinnan suorittajaa ja ohjelmoijana käytännön ohjelmointityön toteuttajaa.<sup>309</sup> Kirjallisuudessa tietokoneohjelman tekijyys on voitu määritellä hieman suppeamminkin katsomalla tekijän olevan käytännössä koodin kirjoittaja<sup>310</sup>. Käytännön ohjelmoija voisi kuitenkin olla vain tekninen avustaja tilanteessa, jossa tekijäksi katsottava henkilö antaisi hänelle niin yksityiskohtaiset ohjeet, ettei koodin kirjoittamiseen jäisi sijaa luovuudelle<sup>311</sup>.

Tekijänoikeusneuvoston lausuntokäytännössä on myös todettu, ettei ohjelman suunnittelun alussa tehtävä ongelmanmäärittely ja -rajaaminen vielä ole tekijän aseman saamiseen johtavaa luovaa toimintaa<sup>312</sup>. Innovaatioprosessin ns. sumeassa alkuvaiheessa (*fuzzy front end*) tapahtuva prosessin suunnan hakeminen sekä segmentin ongelmien ja tarpeiden hahmottelu, johon iteraation ensimmäiset kokeilut voivat liittyä, asettunevat tähän tekijänoikeusneuvoston tarkoittamaan tekijänoikeuden ulkopuolelle jäävän työskentelyn

308 Takki 2003, s. 61–62

309 Ks. esim. TN 1989:7, TN 1993:12, TN 2003:10 ja TN 2007:3

310 Ks. esim. Innanen – Jäske 2014, s. 147 ja Välimäki 2009, s. 26

311 Koivumaa 1995, s. 130

312 Ks. TN 1989:7 ja TN 1993:12

vaiheeseen. Kun kehitettävä idea alkaa kokeilujen myötä hahmottua, muuttuu prototyyppikin yksityiskohtaisemmaksi ja kokeilun tarkoituksiksi tulee pikemminkin tuotteen hienosäätäminen kuin abstraktisten ideakokonaisuuksien hahmottaminen<sup>313</sup>. Voitaisiin ajatella, että prosessin alkupään ideatasoisesta työskentelystä siirrytään kokeilusykli kerhallaan kohti ilmaisumuodon tasolla tapahtuvaa työskentelyä, jossa pyritään toteuttamaan konkreettisia ratkaisuja. Näin ollen teokseksi katsottavia henkisiä tuotoksia syntyisi todennäköisemmin kokeiluprosessin loppu- kuin alkupäässä.

Idean ja ilmaisumuodon tasolla työskentelyn välistä rajaa etsittiin lausunnossa TN 2007:3, jossa ohjelmiston opinnäytetyönään suunnitelleilla hakijoilla ei katsottu olevan tekijänoikeutta itse tietokoneohjelmaan, vaan ohjelman tekijänä pidettiin käytännön ohjelmointityön suorittajaa. Opinnäytetyön tekijät A.A. ja N.N. suunnittelivat ja ideoivat ohjelmiston, tekivät vaatimusmäärittelyjä, hyväksyivät ohjelmoijan laatimat ehdotukset ja laativat ohjelmiston sisältämän kirjallisen materiaalin. Vaikka hakijat olivat vastanneet järjestelmän ideoinnista ja projektin kuvailtiin tapahtuneen hakijoiden ja ohjelmointityön suorittajan tiiviinä yhteistyönä, ei tietokoneohjelma neuvoston näkemyksen mukaan ollut yhteisteos. Sen sijaan ohjelmoijan luoma ohjelmakoodi ja hakijoiden ohjelmistoon tuottama teksti muodostivat yhdessä kokoomateoksen, jonka hyödyntäminen kokonaisuutena edellytti jokaisen tekijän suostumusta.

Ryhmätyön ja koehenkilöiden osallistamisen ohella kokeiluille on tyypillistä prosessin syklinen eteneminen. Kun ideaa kehitetään peräkkäin toistettavissa innovaationsykleissä, joissa aiemmin luotua prototyyppiä hiotaan kohti paremmin toimivaa tuotetta, on kyse yhteisteoksen luomisesta vertikaalisen yhteistyön muodossa, jos luomistyöhön osallistuvien henkilöiden piiri prosessin edetessä vaihtelee.

### **6.2.3 Voiko koehenkilö olla tekijä?**

Kysymys koehenkilön tekijänoikeudellisesta roolista kokeilussa riippuu siitä, voidaanko koehenkilön toiminta katsoa sellaiseksi luovaksi työskentelyksi, jonka lopputuloksena syntyy teos. Kovin yleispäteviä periaatteita koehenkilön tekijänoikeudellisesta roolista on kokeilutilanteiden erilaisuuden<sup>314</sup> vuoksi mahdoton johtaa, joten lopulta koehenkilön asema ratkeaa kussakin yksittäistapauksessa hänen suorittamiensa toimien perusteella.

---

<sup>313</sup> Hassi ym. 2015, s. 50

<sup>314</sup> Erilaisista kokeiluasetelmista ks. 2.3.3

Jos koehenkilö osallistuu kokeiluun lähinnä antamalla sanallisesti palautetta ja parannusehdotuksia, hänen osallistumistaan tuskin voidaan ymmärtää sellaiseksi ohjelmiston syntyyn vaikuttavaksi luovaksi toiminnaksi, joka voisi antaa tekijänoikeuden johonkin ohjelmiston osaan. Pääsääntöisesti idean konkretisointi ja lopullisten ratkaisujen tekeminen kuuluvat kehitystiimille, joka luo prototyypit ja päättää niihin tehtävistä muutoksista. Esimerkiksi lausunnossa TN 1989:7 hakijoiden ei katsottu osallistuneen konekirjoituksen opetukseen liittyvän tietokoneohjelman luomiseen, koska heidän panoksensa oli rajoittunut ohjelman ideoimiseen, kokeilemiseen ja palautteen antamiseen pedagogisen asiantuntijan roolissa. Samanlainen päätelmä voitaneen tehdä tilanteissa, joissa koehenkilö toimii eräänlaisena käyttäjäasiantuntijana, joka vain kokeilee tuotetta ja antaa sen toimivuudesta palautetta.

Ohjelmistoalan kokeilussa tekijänoikeudella suojattu tietokoneohjelma eli ohjelman koodi tuskin missään vaiheessa ilmenee koehenkilölle – eikä maallikko edes ymmärtäisi koodia, vaikka siihen käsiksi pääsisikin. Ohjelmistoa koskevia kokeiluja voidaan suorittaa myös ilman koodin kirjoittamista esimerkiksi paperiprototyypin, digitaalisten työkalujen tai manuaalisten simulointien avulla, joten tällaisessakaan kokeilussa kokeiluun osallistuville ei voi muodostua tekijänoikeutta itse koodiin. Koehenkilön tuleminen ohjelmakoodin tekijäksi edellyttäisi henkilöltä jonkinlaista ohjelmointia koskevaa asiantuntemusta, pääsyä kehitettävän ohjelman lähdekoodiin sekä osallistumista lähdekoodin tuottamiseen. Vaihtoehtoisesti hänen tulisi sanella ohjelmoijalle koodi niin tarkasti, että ohjelmoija katsottaisiin vain tekniseksi avustajaksi. Tavallisesti koehenkilöltä tuskin odotetaan tällaista panosta.

Koska koehenkilön on kokeilussa tarkoitus edustaa tuotteen käyttäjän näkökulmaa, hän luultavasti tarkastelee ohjelmistoa pikemminkin käyttöliittymän ja toiminnallisuuksien kuin lähdekoodin tasolla. Koodin ohella ohjelmistoon voi kuulua muitakin teostasoon yltäviä elementtejä, joten koehenkilölle voisi syntyä tekijänoikeus johonkin ohjelmiston osaan, vaikka hän ei pääsisi käsiksi tietokoneohjelman koodiin. Tällöin ohjelmiston hyödyntäminen vaatisi myös koehenkilön luvan. Esimerkiksi lausunnossa TN 2007:3 ohjelmoijan luoman tietokoneohjelman ja hakijoiden ohjelmistoon toteuttaman tekstin katsottiin molempien yltävän teostasoon ja muodostavan yhdessä kokoomateos, jota ei kokonaisuutena voitu luovuttaa edelleen ilman kaikkien tekijöiden suostumusta.

Yksinomaan sanallisesti viestien koehenkilö voisi lähinnä tulla jonkin ohjelmistoon kuuluvan kirjallisen teoksen tekijäksi, jos jotain hänen kielellisesti ilmaisemaansa sisältöä



sanatarkasti liitettäisiin ohjelmistoon. Sen sijaan jonkin koehenkilön kielellisesti kuvailman asian sisällyttäminen ohjelmistoon vaikkapa kuvateoksena (esim. käyttöliittymän graafisena elementtinä) tuskin muodostaisi tekijänoikeuden loukkausta, koska verbaalinen viesti ei ole suoraan (ts. epäitsenäisesti ja vailla minkäänlaista omaperäistä ajattelua) käännettävissä visuaaliseksi ilmaisuksi. Koska kokeiluissa painotetaan käytännön toimintaa pelkän verbaalisen viestinnän sijasta, ei välttämättä olisi täysin poissuljettua ajatella, että koehenkilölle annettaisiin mahdollisuus myös osallistua aktiivisesti prototyypin muokkaamiseen. Tällaisessa tapauksessa koehenkilö saattaisi pelkän ideatasoisen työskentelyn sijasta osallistua myös teoksen tekijänoikeuden alaisuuteen kuuluvan ilmaismuodon luomiseen.

## 6.3 Oikeuksien siirtyminen

### ***6.3.1 Kokeiluun osallistuvien henkilöiden ja yrityksen välinen oikeussuhde***

Jos ohjelmistokehitystä tekevä yritys toteuttaa kokeilunsa itse (sen sijaan, että antaisi kokeilun muun tahon tehtäväksi ja pyrkisi vain hyödyntämään kokeilun tuloksia), kokeiluun ohjaava kehitystiimi koostunee yritykseen työsuhteessa tai muunlaisessa toimeksiantosuhteessa olevista henkilöistä. Heidän lisäksi kokeiluun osallistuu koehenkilöitä eli yrityksen ulkopuolelta tulevia käyttäjiä. Kokeiluun osallistuvien henkilöiden ja yrityksen välisestä oikeussuhteesta riippuu, tarvitaanko tekijänoikeuksien siirtämiseen yritykselle erityistä sopimusta vai voisiko tekijänoikeus siirtyä yritykselle suoraan lain tai oikeuskäytännössä muotoutuneen normaalikäytösäännön nojalla.

Erityisesti ohjelmistoyrityksen ja kokeiluun koehenkilön roolissa osallistuvien henkilöiden välisen oikeussuhteen laatu voi olla epäselvä, jos nimenomaista sopimusta ei ole solmittu. Vedenkannas esittää living lab -ympäristön oikeudellisia reunaehdoja koskevassa artikkelissaan, että living labin ja käyttäjien välillä voitaisiin nimenomaisen sopimuksen puuttuessa katsoa vallitsevan joko oikeudellisesti sitomaton moraalinen sopimus tai konkludenttinen eli hiljainen juridinen sopimus. Konkludenttinen sopimus voisi olla tulkittavissa myös työ sopimukseksi.<sup>315</sup> Johtopäätelmän voitaisiin kokeilu- ja living lab -toi-

---

315 Vedenkannas 2010, s. 854–856



minnan yhtenevien piirteiden<sup>316</sup> vuoksi nähdä analogisesti soveltuvan joissain tilanteissa myös kokeiluun osallistuvaan koehenkilöön, joka siten saattaisi olla kokeilusta vastaavan yrityksen työntekijä työsopimuslain (TSL, 26.1.2001/55) edellyttämien tunnusmerkkien täytyessä.

Työsopimuslain soveltuvuus koehenkilön ja yrityksen oikeussuhteeseen ratkeaa TSL 1:1.1 §:ssä ilmaistujen työsuhteen perustunnusmerkkien nojalla. Lainkohta pitää sisällään sopimus-, työntekemis-, vastikkeellisuus- ja direktiotunnusmerkin. Ensinnäkin työsuhteen tulee siis perustua sopimukseen, joka tosin ei edellytä tiettyä määrämuotoa, vaan sopimus voi perustua työnantajan hiljaiseen suostumukseenkin. Toiseksi työntekijän tulee sopimuksella sitoutua tekemään työtä työnantajan lukuun eli työstä saatavan hyödyn on koiduttava työnantajan hyväksi. Työn käsite on siten laaja, että se voi olla mitä vain ihmillistä käyttäytymistä ja toimintaa, jolla on taloudellista merkitystä. Vastikkeellisuustunnusmerkki tarkoittaa sitä, että työntekijä sitoutuu tekemään työtä vastiketta vastaan. Vastikkeen laatua ei ole TSL:ssa tarkemmin määritelty, vaan se voi olla yhtä hyvin rahaa kuin tavaraakin, kunhan sillä on saajalleen taloudellista arvoa. Neljännen eli direktiotunnusmerkin mukaisesti työsuhteeseen kuuluu työnantajan oikeus johtaa työntekoa ja valvoa työnteon laatua. Em. tunnusmerkkien tarkastelun ohella suoritetaan yksittäistapauksessa kokonaisarviointi, jossa voidaan huomioida muun muassa sopimuksesta ilmenevä osapuolten tarkoitus.<sup>317</sup>

Living lab -ympäristössä Vedenkannas pitää vastikkeellisuus- ja direktiotunnusmerkkejä ongelmallisimpina. Hän näkee esimerkiksi testaamisen paikan ja ajan määräämisen tai käyttäjälle asetetun raporttoimisvelvollisuuden voivan viitata direktiotunnusmerkin täyttymiseen, kun taas vastikkeellisuuden vaatimus edellyttäisi esimerkiksi jonkin hyödykkeen käyttöoikeuden antamista koekäyttäjälle.<sup>318</sup> Kyseisten tunnusmerkkien toteutuminen on kokeilujenkin kohdalla epävarmaa ja riippuu suuresti kokeilun toteuttamistavasta. Kokeilu voidaan suunnitella niin, ettei kehitystiimillä ole juuri mahdollisuutta ohjata ja valvoa koehenkilöiden toimintaa tuotteen käyttämisen aikana, vaan kokeilun hyöty voi

316 Vedenkannaksen 2010, s. 847, 860–861, mukaan living lab -toiminnassa käyttäjiä osallistetaan T&K-toimintaan siten, että käyttäjät käyttävät tuotteita tai palveluita osana arkielämäänsä ja antavat kokemuksestaan palautetta toimeksiantajalle. Näin käyttäjän rooli tuotekehitysprosessissa on varsin samankaltainen kuin kokeiluun osallistuvan koehenkilön. Living labin käsite näyttäisi viittaavan ennen kaikkea useiden toimijoiden muodostamaan vuorovaikutusympäristöön, jossa T&K-toiminta tapahtuu (ks. esim. Heikkanen – Österberg 2012, s. 10), kun taas kokeilu on käyttäjiä osallistava menetelmä, jota voitaisiin hyödyntää myös living lab -ekosysteemeissä. Living lab -casen ja kokeilun toteutusvaihe ovat molemmat käyttäjäkeskeisiä ja osallistavia prosesseja, jotka etenevät iteraatiosykleissä, mutta kokeilussa keskeistä on lisäksi syklien nopeus ja edullisuus. Ks. living lab -prosessin toteutusvaiheesta esim. Heikkanen – Österberg 2012, s. 42–43.

317 Rautiainen – Äimälä 2008, s. 18–23 ja Tiitinen – Kröger 2012, s. 9–19, 107–108

318 Vedenkannas 2010, s. 856

perustua vain koehenkilöiden antamaan palautteeseen. Direktiotunnusmerkin täyttymistä vastaan voisi puhua seikin seikka, että koehenkilö osallistuu kokeiluun vapaaehtoisesti ja vain siinä määrin kuin itse tahtoo. Vastikkeellisuustunnusmerkin osalta voidaan todeta, ettei tuotekehitykseen osallistuvaa käyttäjää tavallisesti tarvitse motivoida tarjoamalla hänelle vastiketta panoksestaan<sup>319</sup>. Usein koehenkilö osallistuu kokeiluun muista syistä kuin ansiotarkoituksessa, eikä täysin ilman ansiotarkoitusta tapahtuva työ tavallisesti ole katsottavissa työsuhteessa tapahtuvaksi<sup>320</sup>. Koehenkilön katsominen työntekijäksi olisi siis teorian tasolla mahdollista, mutta tyypillisesti kokeilussa kaikki työsuhteen tunnusmerkit tuskin täyttyvät.

### **6.3.2 Oikeuksien siirtäminen sopimalla**

Tekijänoikeuden siirtämisestä työsuhteen ulkopuolella on aina sovittava, sillä ilman työsuhdetta oikeudet eivät siirry tekijänoikeuslain tai työsuhteisiin sovellettavan normaalkäytösäännön nojalla. Ohjenuora pätee myös erilaisten toimeksiantosuhteiden kohdalla, kuten tapauksissa, joissa teoksen luomiseen osallistuu konsultti, alihankkija tai muu sopimuskumppani: tällöinkin oikeuksien siirtyminen edellyttää sopimusta.<sup>321</sup>

Tekijänoikeuslaki perustuu sopimusvapaudelle, ja tekijänoikeus on lähtökohtaisesti luovutettavissa. TekijäL 27.1 §:n mukaan tekijä voi luovuttaa oikeutensa kokonaan tai osittain, joskaan hän ei voi luopua moraalisisista oikeuksistaan kuin 3.3 §:n asettamissa rajoissa. Myös tulevien teosten tekijänoikeuksien luovuttamisesta voidaan ennakoon sopia. TekijäL 3 luku pitää sisällään yleiset säännökset tekijänoikeuden siirtymisestä, mutta 27.3 §:n mukaan 30–40 sekä 40 b § ovat tahdonvaltaisia eli niitä sovelletaan vain, elleivät osapuolet ole tekijänoikeuden luovuttamisesta muuta sopineet. Tekijänoikeuksien siirtoa koskevalta sopimukselta ei edellytetä tiettyä määrämuotoa, joten myös suullinen tai jopa konkludenttinen sopimus on pätevä (joskin voi aiheuttaa ongelmia näytön ja tulkinnan suhteen).<sup>322</sup>

TekijäL 28 §:n mukaan tekijänoikeuden luovutuksensaaja ei saa ilman erillistä sopimusta muuttaa teosta tai luovuttaa sitä edelleen. Sopimus on tässäkin tapauksessa muotovapaa, ja usein esimerkiksi muuttamisoikeudesta voidaan katsoa hiljaisesti sovitun olosuhteiden,

319 Palfrey 2012, s. 60–61, 70. Palfreyn mukaan käyttäjät ovat yhä kiinnostuneempia osallistumaan heille tärkeiden tuotteiden kehitykseen, sillä he hyötyvät osallistumisestaan saamalla tuotteisiin toivomiaan ominaisuuksia. Lisäksi osallistuminen voi olla käyttäjille palkitsevaa siksi, että he pääsevät käyttämään luovuuttaan.

320 Rautiainen – Äimälä 2008, s. 18

321 Takki 2003, s. 54

322 Harenko ym. 2016, s. 334, 343–344 ja Mansala 2007, s. 597

alan tapojen tai työsuhteen tarkoituksen perusteella<sup>323</sup>. Ylipäänsä tekijänoikeuden luovutussopimusta on kuitenkin tulkittava esitöiden ja oikeuskäytännön mukaan suppeasti<sup>324</sup>. Oikeuskirjallisuudessa on tosin esitetty, ettei periaatetta tulisi yksiselitteisesti noudattaa tapauksissa, joissa tekijä ei ole heikompi sopijapuoli<sup>325</sup>.

### **6.3.3 Oikeuksien siirtyminen työsuhteen perusteella**

Työsuhteessa luotujen teosten tekijänoikeuden siirtymisestä ei ole patentteja koskevan oikeudesta työntekijän tekemiin keksintöihin annetun lain (29.12.1967/656) kaltaista sääntelyä tekijänoikeuslaissa tai muuallakaan lainsäädännössä. Poikkeuksen muodostaa TekijäL 40 b §, joka säätelee tietokoneohjelman ja tietokannan tekijänoikeuden siirtymistä työnantajalle, kun teos on luotu työ- tai virkasuhteessa. Pykälän 2 momentin mukaan säännöstä ei kuitenkaan sovelleta korkeakoulun (pl. sotilasopetuslaitokset) opetus- tai tutkimustyössä itsenäisesti toimivan tekijän luomaan tietokoneohjelmaan. Tekijänoikeuslain esitöissä todetaan lainkohdassa mainitulla työsuhteella tarkoitettavan TSL 1:1 §:ssä tarkoitettuun työsopimukseen perustuvaa työsuhdetta<sup>326</sup>, joten TekijäL 40 b §:n sovellettavuus ratkeaa edellä (6.3.1) käsiteltyjen työsuhteen tunnusmerkkien täyttymisen perusteella. 40 b §:ää sovelletaan vain, jos osapuolet eivät ole muuta sopineet (TekijäL 27.3 §).

TekijäL 40 b §:n mukaan tekijänoikeus siirtyy työnantajalle silloin, kun teos on luotu täytettäessä työsuhteesta johtuvia tehtäviä. Jotta työntekijän tekijänoikeus ohjelmaan siirtyisi suoraan tekijänoikeuslain nojalla työnantajalle, tulee tietokoneohjelmien luomisen siis kuulua tekijän normaaliin toimenkuvaan<sup>327</sup>. Haarmann ja Mansala näkevät 40 b §:n voivan aiheuttaa tulkintavaikeuksia silloin, kun tietokoneohjelman tekijän varsinaisiin työtehtäviin ei kuulu ohjelmien luominen<sup>328</sup>. Kysymystä sivuttiin lausunnossa TN 1993:12, jossa oli kyse kaupungin liikunnanohjaajan tekijänoikeudesta tietokoneohjelmaan, jonka kehittämiseen hän oli osallistunut. Koska lausuntopyyntöön vastaaminen olisi vaatinut tosi-seikkojen selvittämistä, näytön arviointia ja sopimusten tulkintaa, ei tekijänoikeusneuvosto voinut juuri ottaa kantaa tapaukseen. Jos liikunnanohjaaja kuitenkin olisi tapauksessa katsottu ohjelman tekijäksi, olisi 40 b §:n sovellettavuuden ratkaisemiseksi tapauksessa

323 Lohi 2009, s. 41

324 KM 1953:5, s. 63. Oikeuskäytännössä suppean tulkinnan periaate on vahvistettu esim. tapauksissa KKO 1984 II 26 sekä KKO 2005:92. Lohi 2009, s. 52, mukaan luovutusta ei kuitenkaan tule tulkita suppeasti, jos luovutus tapahtuu työsuhteessa ja teoksen käyttö kuuluu työnantajan normaalin toiminnan piiriin. Tällöin oikeudet siirtyvät normaalikäytösäännön osoittamassa laajuudessa.

325 Harenko ym. 2016, s. 342

326 HE 161/90 vp., s. 53

327 HE 161/90 vp., s. 53–54

328 Haarmann – Mansala 2012, s. 146

jouduttu arvioimaan sitä, kuuluiko tietokoneohjelman luominen liikunnanohjaajan työtehtäviin ja oliko ohjelma luotu työajalla.

Tekijäl 40 b § koskee tietokoneohjelmia ja siihen välittömästi liittyviä teoksia sekä tietokantoja. Koska tietokoneohjelmien tekijänoikeussuoja kattaa myös ohjelman teostasoon yltävän valmisteluaineiston, siirtyy tekijänoikeus tällaiseen valmisteluaineistoon 40 b §:n nojalla samalla lailla kuin ohjelman valmiin versionkin tekijänoikeus<sup>329</sup>. Ohjelmaan välittömästi liittyvinä teoksina mainitaan esitöissä ohjelmakuvaukset ja tukimateriaali, mutta oikeuskirjallisuudessa tällaisiksi 40 b §:n mukaisesti siirtyviksi teoksiksi on katsottu myös ohjelmaan sisältyvät tekstitiedostot, ohjetekstit sekä käyttöliittymän grafiikat<sup>330</sup>. Vaikka mm. ohjelman kehittämiseen liittyvää esitutkimusta, vaatimusmäärittelyjä ja toiminnallisia määrittelyjä ei pidetä tietokoneohjelman ilmenemismuotona, voivat oikeudet niihin siirtyä työnantajalle Tekijäl 40 b §:n nojalla, jos ne katsotaan ohjelmaan välittömästi liittyviksi teoksiksi<sup>331</sup>.

Suoraan tekijänoikeuslain nojalla tapahtuva tietokoneohjelman oikeuksien siirtyminen kattaa Tekijäl 2 §:n mukaiset taloudelliset oikeudet<sup>332</sup>. Siirtyvien oikeuksien laajuuden on mahdollistettava työnantajan tavanomaisen liiketoiminnan harjoittaminen<sup>333</sup>. Tekijänoikeus siirtyy työnantajalle pysyvästi, eli oikeus ei palaudu työntekijälle työsuhteen päättyessä<sup>334</sup>. Harenko ym. pitävät toistaiseksi epäselvänä sitä, antaako Tekijäl 40 b § työnantajalle yksinomaisen oikeuden ohjelmaan ja onko työnantajalla oikeus myös luovuttaa ohjelman oikeudet edelleen ja muunnella ohjelmaa Tekijäl 28 §:n tarkoittamalla tavalla. Ohjelmistodirektiivin 2 art. 3 kohdan voitaisiin tulkita viittaavan siihen, että työnantaja saisi myös edelleenluovutus- ja muunteluoikeuden, mutta kantaa ei ole oikeuskäytännössä vahvistettu. Ohjelmistoalalla vallitsevana tapana on kuitenkin muuttamisoikeuden laaja siirtyminen työnantajalle työsopimussuhteen perusteella<sup>335</sup>. Myös komiteanmietinnössä 1987:8 kyseisen käytännön katsotaan olevan siinä määrin vakiintunut, että ohjelmoijan voitaisiin yleensä katsoa hiljaisesti antaneen suostumuksensa muutosten tekemiseen ilman nimenomaista muuttamisoikeuden luovuttamistakin<sup>336</sup>.

329 HE 211/92, s. 3

330 HE 161/90 vp., s. 54, Harenko ym. 2016, s. 405 ja Takki 2003, s. 54. Vrt. Välimäki 2009, s. 31, jonka mukaan 40 b § ei kata tietokoneohjelmaan liittyviä osia, kuten kuvateoksia ja tekstiä, joiden siirtymisestä olisi Välimäen mukaan sovittava erikseen.

331 Innanen – Jäske 2014, s. 148–149

332 HE 161/90 vp., s. 54

333 Harenko ym. 2017, s. 406

334 HE 161/90 vp., s. 54

335 Harenko ym. 2017, s. 352, 406. Vrt. TN 2007:3, jossa Tekijäl 28 §:n ei katsottu koskevan Tekijäl 40 b §:n tarkoittamia tapauksia, vaan ainoastaan sopimukseen perustuvia tekijänoikeuden luovutuksia. Tekijänoikeusneuvosto perusteli tulkinnan ohjelmistodirektiivin 2 art. 3 kohdalla, ohjelmistoliiketoiminnan tarkoituksella sekä sillä, että Tekijäl 28 § on säädetty aikana ennen tietokoneohjelmia koskevan sääntelyn ottamista tekijänoikeuslakiin.

336 KM 1987:8, s. 189

TekijäL 40 b § pitkälti estää tietokoneohjelmien oikeuksia koskevat epäselvyydet tapauksissa, joissa ohjelman tekijät ovat sellaisia yrityksen työntekijöitä, joiden normaaliin toimenkuvaan ohjelmien luominen kuuluu. Oikeuksien siirtyminen suoraan lain nojalla poistaa tarpeen sopia jokaisen työntekijän kanssa nimenomaisesti oikeuksien siirtämisestä, mutta se myös helpottaa yritysten sekä yritysten ja tutkimuslaitosten välisiä tuotekehitysprojekteja<sup>337</sup>. TekijäL 40 b § ei kuitenkaan turvaa yrityksen oikeutta hyödyntää teosta, jos ohjelmiston luomiseen osallistuu muitakin kuin yrityksen työntekijöitä, joille voi syntyä tekijänoikeus johonkin ohjelmiston osaan.

Jos TekijäL 40 b § ei sovellu, mutta teos on luotu työsuhteessa, voi työnantaja saada käyttöoikeuden teokseen oikeuskäytännössä muotoutuneen normaalikäytösäännön perusteella. Normaalikäytösäännön mukaan työnantaja saa ilman nimenomaista sopimustakin työntekijän luomaan teokseen normaalin toimintansa edellyttämän käyttöoikeuden.<sup>338</sup> Usein tämä antaa työnantajalle kuitenkin vain ”kertakäyttöoikeuden” ja laajempi käyttöoikeus on mahdollista saada vain asiasta tekijän kanssa erikseen sopimalla<sup>339</sup>.

## 6.4 Strategisia näkökohtia

Yrityksen piirissä syntyneiden immateriaalioikeuksien on suositeltavaa olla kokonaan yrityksen, eikä esimerkiksi yrityksen työntekijöiden tai perustajien, nimissä<sup>340</sup>. Jos tekijänoikeutta ei siirretä yritykselle, on yrityksen saatava tekijältä lupa teoksen TekijäL 2 §:n mukaiseen hyödyntämiseen. Erityisesti silloin, kun teoksella on useita tekijöitä, voi teoksen hyödyntäminen olla kankeaa, koska se vaatii kaikkien tekijöiden suostumuksen.

Koska kokeilu voidaan toteuttaa monin eri tavoin, on mahdoton hahmottaa joka tilanteessa päteviä periaatteita sen suhteen, millaisiksi koehenkilön ja kehitystiimin tekijänoikeudelliset asemat muotoutuvat ja miltä osin kokeilussa syntyvän teoksen tekijänoikeudet siirtyvät kokeilusta vastaavalle yritykselle. Ohjelmistotuotannossa tilanne on tekijänoikeuksien siirtymisen kannalta hieman muita toimialoja yksinkertaisempi, sillä oikeudet työntekijän kehittämään tietokoneohjelmaan ja siihen välittömästi liittyvään teokseen siirtyvät TekijäL 40 b §:n nojalla työnantajalle. Varmuuden vuoksi oikeuksien siirrosta on

337 Harenko ym. 2017, s. 405

338 Lohi 2009, s. 40

339 Haarmann – Mansala 2012, s. 145

340 Honkinen – Innanen – Lindgren – Pello – Rantanen – Siltala – Tuomala 2016, s. 111

silti tietokoneohjelmia kehittävien henkilöidenkin kanssa suositeltavaa nimenomaisesti sopia esim. työ sopimuksessa<sup>341</sup>. Oikeuskirjallisuudessa ei ole täysin yksiselitteisesti määritelty, mitkä kaikki ohjelmiston elementit voidaan katsoa TekijäL 40 b §:ssä tarkoitetuiksi tietokoneohjelmaan välittömästi liittyviksi teoksiksi, ja tulkintaongelmia voi syntyä myös tapauksissa, joissa ohjelman luominen ei ole kuulunut työntekijän normaaleihin työtehtäviin. Tällöin normaalikäytösäännön tarjoamasta ”varaventiilistä” huolimatta nimenomainen ennakkoon sopiminen kannattaa myös työntekijöiden kanssa.

Tekijänoikeudet eivät siirry lain tai normaalikäytösäännön perusteella, jos tekijä ei ole työsuhteessa yritykseen (esim. koehenkilö, jonka kohdalla työsuhteen tunnusmerkit eivät täyty). Yrityksen on erikseen sovittava oikeuksien siirtämisestä tällaisten henkilöiden kanssa, jos heille voi kokeilussa syntyä tekijänoikeus ohjelmistoon. Huomionarvoista on myös, ettei tietokoneohjelma ole ohjelmiston ainoa mahdollisesti tekijänoikeussuojaa saava osa, vaan ohjelmistoon voi kuulua muitakin teoksia, joiden osalta tarvittavien oikeuksien siirtämisestä on huolehdittava.

Koehenkilön roolin ennakoimista voi hankaloittaa se, että kokeilussa ei suunnitteluun perustuvien prosessien lailla tiedetä, mitä kokeilun edetessä tulee tapahtumaan. Kokeilun iteratiivisen luonteen vuoksi tarkkaa suunnitelmaa voi olla vaikea ennalta hahmotella. Kokeilun suunnitteluvaiheessa on kuitenkin mahdollista määrittää, millaisella kokeiluasetelmalla ja prototyypillä kehitettävää ideaa lähdetään testaamaan – ja näin vaikuttaa siihen, millaisessa vuorovaikutussuhteessa koehenkilö on suhteessa prototyyppiin ja millaisia tehtäviä hänelle annetaan. Jos koehenkilö osallistetaan kokeiluun vain sanallista palautetta antavana ohjelmiston koekäyttäjänä, joka ei kokeilun kestäessä voi luoda tai muokata mitään ohjelmiston osaa, koehenkilölle tuskin voi syntyä tekijänoikeutta. Koehenkilön ja kehitystiimin välillä ei kaikissa tapauksissa ole edes tällaista vuorovaikutusta, vaan käyttäjien kiinnostusta ohjelmistoa kohtaan voidaan testata esimerkiksi seuraamalla tuotteen jonotuslistalle ilmoittautuneiden henkilöiden määrää<sup>342</sup>. Näin ollen tekijyyttä ja tekijänoikeuksien siirtymistä koskevilta ongelmatilanteilta voitaisiin välttyä määrittelemällä koehenkilön rooli ennalta sellaiseksi, että hänellä ei ole mahdollisuutta osallistua teoksen ilmaisumuodon luomiseen. Pääsääntöisesti koehenkilön kokeiluun antama panos lieneekin katsottavissa vain ideoiksi, jotka eivät voi saada tekijänoikeussuojaa.

341 Innanen – Jäske 2014, s. 141 ja Mansala 2001, s. 14

342 Näin toimittiin esimerkiksi tiedostojen jakamiseen tarkoitetun Dropboxin kohdalla. Ks. Ries 2016, s. 117–119.

Jos koehenkilö kuitenkin halutaan osallistaa tuotekehitysprosessiin tiiviimmin ja häneltä jopa toivotaan jonkinlaista luovaa panosta, voi tekijänoikeuden syntyminen koehenkilölle olla ainakin teoriassa mahdollista. Vedenkannas suosittelee living lab -toimintaan yksinkertaisia vakiosopimuksia labin ja testikäyttäjien välille epäselvyyksien välttämiseksi. Sopimuksessa ilmaistaisiin labin ja käyttäjän välisen oikeussuhteen ehdot, kuten toiminnassa syntyvien immateriaalioikeuksien kuuluminen labille.<sup>343</sup> Tällainen vakiosopimus voisi luoda varmuutta myös sellaisessa kokeilussa, jossa koehenkilölle annettaisiin tavallista aktiivisempi rooli. Yksilöllisten sopimusten neuvottelemisen kunkin koehenkilön kanssa ei liene tarkoituksenmukaista, vaan se voisi pikemminkin olla vastoin kokeilun nopeudelle ja edullisuudelle perustuvaa luonnetta sekä toisaalta antaa myös koehenkilölle prosessista raskaan ja epämotivoivan vaikutelman.

Kirjallisen sopimisen tärkeyttä kokeiluun osallistuvien henkilöiden kanssa korostaa se, että sopimuksen perusteella saamaansa oikeuteen vetoavalla on sopimuksen olemassaolosta näyttötaakka. Tämä pätee myös työsopimussuhteen olemassaolon todisteluun, kun tekijänoikeuden väitetään siirtyneen työnantajalle TekijäL 40 b §:n nojalla.<sup>344</sup> Niin tekijänoikeuksien luovuttamista kuin työsuhteen solmimistakin koskevat sopimukset kannattaa siis laatia kirjallisena näyttövaikeuksien välttämiseksi. Luovutus sopimusten suppean tulokinnan vuoksi sopimuksessa kannattaa myös nimenomaisesti ilmaista kaikki oikeudet, jotka yritys tarvitsee ohjelmiston hyödyntämiseksi.

---

343 Vedenkannas 2010, s. 860

344 Harenko ym. 2016, s. 335–336



## 7. TEKIJÄNOIKEUS, MUUT SUOJAKEINOT JA NIIDEN YHTEISKÄYTTÖ

Vaikka tekijänoikeutta on perinteisesti pidetty tietokoneohjelmien kannalta tärkeimpänä suojakeinona, on myös patenttien merkitys ohjelmistoalalla kasvanut. Läheskään kaikki yritykset eivät kuitenkaan hyödynnä – ainakaan tietoisesti – mitään oikeudellisia suojakeinoja, vaan etenkin pienet ja nopeasti kehittyvillä aloilla toimivat yritykset turvautuvat tyypillisesti lakiin perustumattomiin keinoihin.<sup>345</sup> Tässä luvussa käsitellään tiiviisti ensin patentteja ja sitten ei-muodollista suojaamista osana kokeilevan ohjelmistoyrityksen strategiaa ja vertaillaan suojakeinoja tekijänoikeuteen. Lopuksi luodaan katsaus avoimeen innovaatioon ja avoimuuden soveltamiseen kokeiluissa.

### 7.1 Patentti

#### **7.1.1 Tietokoneohjelmalla toteutetun ratkaisun patentoitavuus**

PatL 1.2 §:n mukaan ”pelkästään tietokoneohjelmaa” ei katsota keksinnöksi, joten pelkkä tietokoneohjelma ei ole myöskään patentoitavissa. Vastaava rajoitus ilmaistaan Euroopan patenttisopimuksen<sup>346</sup> 52 artiklassa. Tietokoneohjelma tai sen osa voi kuitenkin toteuttaa patenttoimiskelpoisen keksinnön, ja ns. tietokonesovelteisia keksintöjä patentoidaankin Euroopassa Euroopan patenttiviraston käytännön ohjaamana jo vakiintuneesti. Patentoitava keksintö voi olla menetelmä tietyn laitteen toimintoja varten, laite tietyn menetelmän toteuttamiseksi tai tietyssä laitteessa jopa itse tietokoneohjelma. Ratkaisevaa on se, millaisen teknisen vaikutuksen tietokoneohjelman käyttäminen aikaansaa.<sup>347</sup> Ohjelmistojen patentoitavuutta on pidetty tarpeellisenä tekijänoikeussuojan ulottuessa lähinnä lähdekoodiin kirjallisena teoksena<sup>348</sup>, mutta ohjelmistopatentteja on myös kritisoitu. Erityisesti avoimen lähdekoodin kannattajat ovat näkyvästi vastustaneet ohjelmistopatentteja.<sup>349</sup>

345 Päällysaho – Kuusisto 2006, s. 29–30

346 Yleissopimus eurooppapatenttien myöntämisestä, SopS 8/1996

347 Oesch – Pihlajamaa – Sunila 2014, s. 89 ja Välimäki 2009, s. 92

348 Ks. esim. Pykälä 2008

349 Ohjelmistopatenttien kritiikistä ks. esim. Evans – Layne-Farrar 2004, s. 23–68, jotka jakavat kritiikin patenttointiprosessin virheitä koskeviin sekä taloustieteellisiin ja patenttien oikeuttamiseen liittyviin argumentteihin. Patentteja on esim. väitetty myönnettävän ratkaisuille, jotka eivät täytä patentoitavuuden edellytyksiä ja niiden on katsottu hankaloittavan standardien luomista. Samoin on kyseenalaistettu, onko patentilla ohjelmistoalalla luovuuteen kannustavaa vaikutusta. Vrt. esim. Abid 2005, s. 842, jonka mukaan ohjelmistopatentista on myös etuja Open Source -liikkeelle: patentin suoja-aika on tekijänoikeutta lyhyempi, patentin vuosimaksujen vuoksi patentteja ei usein pidetä voimassa täyttä suoja-aikaa ja patenttien rekisteröinti auttaa selvittämään, mihin omaisuuteen liittyy immateriaalioikeuksia ja mikä kuuluu public domainiin.



### 7.1.2 Keksinnön uutuuden säilyttäminen kokeilussa

Patenttilaki ei määrittele keksintöä, jolle patentti voidaan myöntää<sup>350</sup>. Oikeuskirjallisuudessa on kuitenkin hahmoteltu tiettyjä patentoitavuuden edellytyksiä. Kyseeseen tulee teknisesti tehokas ja luonteeltaan tekninen keksintö, joka on toisinnettavissa ja teollisesti hyödynnettävissä. Ratkaisun on oltava uusi ja keksinnöllinen, eikä se saa kuulua patenttilaissa patentoitavuuden ulkopuolelle suljettuihin kohteisiin.<sup>351</sup> Kokeilujen kannalta kriittinen patentoitavuuden edellytys on uutuus, sillä testattavan tuotteen luovuttaminen koehenkilöiden tai aitojen asiakkaiden käytettäväksi voi johtaa tuotteen sisältämän keksinnön uutuuden menettämiseen. Siinä missä tekijänoikeussuojan saamiseen riittää subjektiivinen uutuus<sup>352</sup>, edellytetään patentoitavalta keksinnöltä objektiivista absoluuttista uutuutta: milloin, missä ja millä tavalla tahansa tapahtunut keksinnön julkiseksi tuleminen muodostaa uutuuden esteen, vaikka keksijä olisikin päätenyt ratkaisuunsa itsenäisesti<sup>353</sup>.

PatL 2.1 §:n mukaan keksinnön on oltava uusi siihen verrattuna, mikä on tullut tunnetuksi ennen patenttihakemuksen tekemispäivää. Lisäksi keksinnön on olennaisesti erottava aiemmista ratkaisuista. 2.2 §:ssä määritellään tunnetuksi kaikki, mikä on tullut julkiseksi. Lainkohdan mukaan julkiseksi tuleminen voi tapahtua kirjoituksen tai esitelmän välityksellä, hyväksikäyttämällä tai muulla tavalla. Myös keksijän itsensä toteuttama keksinnön julkistaminen merkitsee PatL 2 §:n tarkoittamaa tunnetuksi tulemistä ja muodostaa siten uutuuden esteen<sup>354</sup>. Tunnetuksi tuleminen ei katso maantieteellisiä rajoja, vaan missä tahansa tapahtunut keksinnön julkistaminen estää sen patentoimisen Suomessa<sup>355</sup>.

Keksintö on tullut julkiseksi, jos ennalta rajoittamattomalla henkilökunnalla on ollut edes teoreettinen mahdollisuus tutustua keksintöön. Esimerkiksi yleisölle avoimessa ympäristössä tapahtuva keksinnön käyttäminen voi tarkoittaa tällaista julkiseksi tulemistä. Keksinnön esitleminen rajatummallekin henkilöpiirille voi tarkoittaa keksinnön tulemistä tunnetuksi, jos keksintöön tutustuneet henkilöt eivät ole velvollisia pitämään keksintöä salassa nimenomaisen sopimuksen tai olosuhteiden perusteella. Jopa yrityksen sisällä tapahtuva keksinnön käyttäminen voi pitemmän ajan saatossa muodostua uutuuden esteeksi, jos henkilökunnan vaihtuvuuden johdosta useat henkilöt pääsevät tutustumaan keksintöön.<sup>356</sup>

350 Haarmann 2014, s. 172

351 Oesch ym. 2014, s. 59

352 Subjektiivisesta uutuudesta tekijänoikeudessa ks. 4.2.1

353 Haarmann 2014, s. 192

354 Oesch ym. 2014, s. 69–70. Eräissä maissa tätä vastoin tunnetaan uutuuden suoja-aika (grace period), jonka kuluessa keksijä voi vielä hakea patenttia, vaikka olisikin julkistanut keksintönsä.

355 Oesch ym. 2014, s. 71

356 Haarmann 2014, s. 192–194 ja Oesch ym. 2014, s. 71–72

Jos yritys siis saattaisi olla halukas patentoimaan kokeilussa testattavan prototyypin tai tuotteen sisältämän keksinnön, olisi patentointimahdollisuudet kyettävä tunnistamaan ennen kokeilun suorittamista. Jotta keksinnön tunnetuksi tuleminen kokeilun toteuttamisen seurauksena ei estäisi keksinnön patentoimista, tulisi patenttihakemus joko jättää ennen keksinnön julkistamista tai vaihtoehtoisesti kokeilu tulisi suunnitella siten, ettei sen suorittaminen merkitsisi keksinnön julkistamista. Uutuuden suojelemiseksi kokeilu voitaisiin ajatella järjestettävän ennalta rajatun, salassapitoon sitoutuneen henkilöjoukon kesken. Alkuvaiheen kokeiluissa, joissa tutkittavia hypoteeseja testataan esim. pelkän kotisivun tai mockupin avulla, patentoitavuuden edellytyksiä tuskin voidaan menettää, sillä luultavasti tällaiset yksinkertaiset prototyypit eivät vielä sisällä patentoitavissa olevia ratkaisuja. Jos kuitenkin yleisölle avoimessa tilassa testataan patenttioikeudellisen keksinnön sisältävää hienostuneempaa prototyyppiä tai jos kokeilu tahdotaan suorittaa päästämällä markkinoille keksinnön sisältävä ns. pienin toimiva tuote, voi yritys samalla menettää mahdollisuutensa ko. ratkaisun patentoimiseen.

### **7.1.3 Tekijänoikeus ja patentti kokeilussa: vertailua**

Keksijälle ei automaattisesti synny yksinoikeutta keksintöönsä (vrt. tekijänoikeus), vaan PatL 1.1 §:ssä ilmaistaan, että keksintöön voidaan myöntää patentti hakemuksesta. Näin ollen patentoitavuuden edellytykset täyttävän keksinnön kohdalla joudutaan aina tekemään valinta sen suhteen, haetaanko keksinnölle patenttia vai ei. Tilanteessa voidaan turvautua kolmeen vaihtoehtoiseen strategiaan: keksinnön patentoimiseen, salaamiseen tai julkistamiseen<sup>357</sup>. Sekä patentointi (jos patenttihakemusta ei ole jätetty ennen kokeilun järjestämistä) että salaaminen strategioina vaativat jonkinlaisen salassapitositoumuksen edellyttämistä kokeiluun osallistuvilta koehenkilöiltä, jos kokeilussa on tarkoitus esitellä tai käyttää keksinnön sisältävää tuotetta. Kokeilua ei myöskään tällöin voida ajatella järjestettävän yleisölle avoimessa tilassa, ennalta rajoittamattoman henkilöjoukon läsnä ollessa tai päästämällä pienin toimiva tuote markkinoille. Sen sijaan em. tavat järjestää kokeilu ovat täysin soveltuvia, jos strategiaksi omaksutaan patentoinnin tai salaamisen sijasta keksinnön julkistaminen. Vain subjektiivista uutuutta edellyttävä tekijänoikeus ei sekään aseta esteitä tuotteen julkistamiselle. Julkistaminen ei anna yritykselle yksinoikeutta keksintöön, mutta keksinnön julkiseksi tulo estää kilpailijoita patentoimasta keksintöä. Vaikka ohjelmistoyritys päättäisi strategiassaan nojautua esimerkiksi julkistamiseen ja nopeisiin markkinaoperaatioihin, eivät sen tuotteet ole täysin vailla oikeudellista suojaa, sillä teostasoon yltävät elementit kuuluvat tällöinkin tekijänoikeuden suojapiiriin.

---

<sup>357</sup> Holgersson – Wallin 2017, s. 1090

Patentin hakeminen on pitkä, kallis ja asiantuntemusta sekä henkilöstöresursseja vaativa prosessi. Yksittäisestä patentista ei myöskään vielä käytännössä usein ole yritykselle juuri hyötyä, vaan patenttien strateginen käyttäminen vaatisi laajempaa patenttiportfolioa, johon pienellä yrityksellä tuskin on edes varaa.<sup>358</sup> Patentointi onkin pk-yrityksissä esimerkiksi tekijänoikeutta harvemmin käytetty suojakeino ja siitä hyötyminen on osoittautunut pienille yrityksille haastavaksi<sup>359</sup>. Brem, Nylund ja Hitchen suosittelevatkin erityisesti pk-yrityksille patentin sijasta muita suojakeinoja, jos yritykset eivät osaa toteuttaa patentointia tehokkaasti<sup>360</sup>. Patentointiprosessin raskauden vuoksi laajamittainen patentointi nopeutta ja edullisuutta tavoittelevassa kokeilussa saattaakin kuulostaa kokeilun tavoitteisiin nähden melko ristiriitaiselta. Tekijänoikeus ja ei-muodolliset suojauskeinot ovat patenttiin verrattuna joustavampia ja halvempia tapoja suojata aineetonta omaisuutta, eikä niiden käyttämiseen liity yhtä paljon byrokratiaa kuin patentointiin. Kuitenkin patentointia lienee syytä harkita laajempaa tulevaisuushorisonttia vasten tarkastellen, eikä pelkästään kokeilua silmällä pitäen. Lisäksi patentoimisen taustalla on usein muitakin syitä kuin pelkkä suojaaminen: patentit voivat auttaa neuvotteluissa, parantaa yrityskuvaa ja houkutella ulkopuolista rahoitusta. Lisäksi patentointiin voivat kannustaa erinäiset sisäiset syyt, kuten työntekijöiden motivointi ja tuotekehityksen tuottavuuden mittaaminen.<sup>361</sup> Tällaisia hyötyjä ei liity ei-muodollisiin suojauskeinoihin, ja tekijänoikeuteenkin vain varsin rajallisesti.

Patentointi voi suojaamisen näkökulmasta olla kannattavaa, jos sillä voidaan suojata jokin arvokas ratkaisu, joka jää tekijänoikeussuojan ulkopuolelle – eli joka käytännössä ilmenee muuten kuin ohjelmakoodin tasolla. Patentti antaa haltijalleen siinä mielessä tekijänoikeutta laajemman suojan, että patentin tuottama yksinoikeus kattaa keksinnön kaikenlaisen kaupallisen käytön, kun taas tekijän yksinoikeudet on tekijänoikeuslaissa hieman tarkkarajaisemmin määritelty. Patentin tuottama yksinoikeus kattaa sellaisetkin erilaiset tietokoneohjelmat, jotka toteuttavat saman keksinnön.<sup>362</sup> Siinä missä tekijänoikeus muoto-idea-dikotomian mukaisesti ulottuu vain teoksen muotoon, voi patentti kattaa sellaisiakin alueita, jotka kyseisen hahmotustavan mukaan jäisivät ideoina suojan ulkopuolelle. Patenttia ja tekijänoikeutta voidaan siis käyttää yhdessä suojaamaan tietokoneohjelman eri ulottuvuuksia. Patentin tuottama kattavampi suoja voi mahdollistaa yritykselle suuremman avoimuuden ja vapaamman tietojen vaihdon ulkopuolisten kanssa ilman riskiä,

358 Välimäki 2009, s. 88–90

359 Holgersson – Wallin 2017, s. 1088 ja Moulin – Thue Lie 2005, s. 4

360 Brem – Nylund – Hitchen 2017, s. 1300

361 Holgersson – Granstrand 2017, s. 1267, 1274

362 Välimäki 2009, s. 89

että ulkopuolinen taho kehittäisi samanlaisen ratkaisun saamansa tiedon pohjalta<sup>363</sup>. Näin ollen patentti voitaisiin nähdä avoimuuden mahdollistavana tekijänä myös kokeilussa, jonka keskiössä on vuorovaikutus yrityksen ulkopuolisten henkilöiden kanssa.

## 7.2 Ei-muodolliset suojauskeinot kokeilussa tekijänoikeuden rinnalla

Immateriaalioikeudet eli muodolliset suojausmenetelmät eivät ole ainoa mahdollisuus suojata aineetonta omaisuutta, ja käytännössä suuri osa yrityksistä suosiikin ei-muodollisia suojausmenetelmiä<sup>364</sup>. Esimerkiksi patentointia useammin tukeudutaan salaamiseen ja nopeaan innovointiin, joskin muodollisten ja ei-muodollisten menetelmien suosio riippuu paljon toimialastakin<sup>365</sup>. Ei-muodollisista suojauskeinoista hieman tarkemmin käsitellään seuraavassa patentoinnille vaihtoehtoisia strategioita (salaaminen ja julkaiseminen) sekä ohjelmistoalalla keskeisenä pidettyä nopeaa innovointitahtia. Muita, tämän esityksen ulkopuolelle jääviä, suojauskeinoja ovat mm. tuotteen monimutkaisuus, tekninen suojaaminen, tehokas tiedon jakaminen, henkilöstön sitouttaminen ja asiakassuhteiden hoitaminen<sup>366</sup>.

Salaaminen on yksi yleisimmin käytetyistä suojauskeinoista, johon usein luotetaan jopa enemmän kuin oikeudelliseen suojaamiseen. Myös tietokoneohjelmien kohdalla salaaminen yhdessä tekijänoikeuden kanssa on perinteisesti ollut merkittävä suojaamismenetelmä: tekijänoikeus on suojannut lähdekoodia, joka on samanaikaisesti pidetty suljettuna eli ohjelmaa on levitetty vain konekielisenä koodina<sup>367</sup>. Kuitenkin salaamisen rinnalle on noussut vaihtoehtoisia strategioita esimerkiksi avoimen lähdekoodin liiketoimintamallien muodossa, ja salaamisen sijasta tuotekehitystä voidaan pyrkiä tehostamaan nimenomaan avoimuudella. Salaaminen voikin myös kokeilussa olla jossain määrin ongelmallista, koska tiedon kulun liiallinen rajoittaminen voi haitata koehenkilöiden kanssa innovointia. Toki mahdollista on rajoittaa kokeiluun osallistuvien koehenkilöiden piiriä ja edellyttää salassapitoon sitoutumista edellä kohdassa 7.1.2 esitetyllä tavalla, jos kehitettävää tuotetta ei toistaiseksi haluta julkistaa.

363 Tarkemmin avoimesta innovaatiosta ks. 7.3.

364 Päällysaho – Kuusisto 2006, s. 15

365 Hall – Helmers – Rogers – Sena 2014, s. 376, 418. Esimerkiksi lääke- ja kemianteollisuudessa patentoinnilla on monia muita aloja tärkeämpi rooli.

366 Päällysaho – Kuusisto 2006, s. 4

367 Hall ym. 2017, s. 387

Ideaa voidaan pyrkiä suojaamaan myös julkaisemalla se. Tämä ehkäisee imitointia kahdella tapaa. Jos kyse on keksinnöstä, merkitsee keksinnön tunnetuksi tekeminen uutuu- den estettä, joka vie pohjan kilpailijoiden mahdollisilta patentoimisaikeilta. Lisäksi idean keksijän tunnettuus voi tehdä kopioimisesta vähemmän houkuttelevaa siksi, että imitoi- jaksi leimautuminen ei ole hyväksi yrityksen maineelle.<sup>368</sup> Kokeilun yhteydessä idean jul- kaiseminen voidaan toteuttaa helposti, jos kokeiluun osallistuvien henkilöiden piiriä ei pidetä suhteellisen rajattuna. Koehenkilöille järjestettävän testaustilanteen ohella kokei- lu voi merkitä myös pienimmän toimivan tuotteen tuomista oikeille markkinoille, mikä luonnollisesti aiheuttaa myös tuotteen julki tuleminen. Tällaisella menettelyllä toteutettu idean varhainen julkistaminen voi estää muita saamasta yksinoikeutta vastaavaan keksin- töön ja samalla toteuttaa avoimuuden periaatteita, mutta toisaalta julkaiseminen merkit- see informaation antamista myös kilpailijoiden käytettäväksi.

Erityisesti ohjelmistoteollisuuden kaltaisilla nopeasti kehittyvillä aloilla keskeinen suo- jaustapa on innovoinnin nopeus, jolla pyritään pitämään etumatkaa kilpailijoihin ja siten estämään imitointia. Yritys siis ehtii jo kehittää tuotteestaan seuraavan version ennen kuin kilpailija on kopioinut sen edellisen tuotteen.<sup>369</sup> Tapauksissa, joissa alan kehitys on ripeää, aikaa ei välttämättä ole käytettävissä esimerkiksi patentin hakemiseen, vaan nopean kas- vun toivotaan kompensoivan suojauksen puutetta. Kuitenkin ihanteellisena vaihtoehtona tunnutaan laajasti pidettävän suojaamista yhdistettynä tehokkaaseen kaupallistamiseen ja nopeisiin markkinaoperaatioihin.<sup>370</sup> Suojauskeinona nopea innovointi vaikuttaisi soveltu- van hyvin yhteen kokeilujen kanssa, sillä kokeiluilla tavoitellaan tehokkaampaa tuoteke- hitystä. Jos kokeiluun onnistutaan nopeuttamaan innovointia, voisi kokeileminen osaltaan jopa tukea tuotteen ei-muodollista suojaamista tuotekehitystahdin ja ensimmäisen lan- seerauksen nopeutumisen muodossa.

Ei-muodollisia suojauskeinoja voidaan pitää varsin joustavina suojaamismenetelmi- nä, sillä keinoja voidaan kehittää yrityksen tarpeita silmällä pitäen – toisin kuin laissa säänneltyjä immateriaalioikeuksia. Ei-muodollisten keinojen käyttöönotto on nopeaa ja helppoa, eikä niiden ylläpitokaan tyypillisesti ole erityisen vaativaa tai kallista.<sup>371</sup> Suojan saamisen nopeuden, helppouden ja edullisuuden suhteen ei-muodollisten suojauskeino- jen ero suhteessa rekisteröimistä vaativiin teollisoikeuksiin on melko selvä, mutta ilman muodollisuuksia syntyvään tekijänoikeuteen nähden ero on pienempi. Vaikka kokeilussa

368 Päällysaho – Kuusisto 2006, s. 16

369 Päällysaho – Kuusisto 2006, s. 20

370 Lahti 2006, s. 142–143

371 Päällysaho – Kuusisto 2006, s. 6, 15, 26

ajan ja varojen rajallisuus eivät puoltaisi patentoimista, eivät kyseiset realiteetit estä tekijänoikeuksien syntymistä. Tekijänoikeuksien strateginen käyttö ei kuitenkaan välttämättä ole täysin yksinkertaista, vaan esimerkiksi Lahti pitää tekijänoikeuksia ennen kaikkea suurten yritysten etuna, sillä tekijänoikeuksien hyödyntäminen liiketoiminnassa vaatii sopimus- ja tekijänoikeusjuridiikan vahvaa osaamista<sup>372</sup>. Toisaalta ei-muodolliset suojakeinotkaan eivät ole aivan ilmaisia, vaan esimerkiksi salaaminen voi aiheuttaa merkittäviäkin kuluja<sup>373</sup>. Lisäksi suojan ylläpitäminen vaatii ei-muodollisten suojakeinojenkin kohdalla aktiivisia toimia<sup>374</sup>.

Tekijänoikeuden ja muiden immateriaalioikeuksien etu ei-muodollisiin suojauskeinoihin nähden on mahdollisuus puolustaa oikeuksia ja puuttua loukkauksiin lain nojalla. Juridisten vaikutustensa lisäksi immateriaalioikeuksiin liittyy myös strategisia etuja, kuten edellä patenttien yhteydessä mainittu mahdollisuus käyttää oikeuksia mm. neuvotteluissa ja innovatiivisen yrityskuvan luomisessa. Tekijänoikeuksilla on rekisteröimättöminä oikeuksina usein tässä mielessä patenttia pienempi merkitys, mutta tekijänoikeuksiakin voidaan hyödyntää esimerkiksi lisensioimalla. Kuitenkin tuotekehitysprosessin varhaisessa vaiheessa, johon ensimmäiset kokeilut sijoittunevat, ei oikeudellinen suoja vielä ehkä tule kyseeseen lain asettamien suojan saamisen edellytysten vuoksi. Ennen idean kaupallisen hyödynnettävyyden selvittämistä kallis rekisteröimisprosessikaan ei välttämättä kannata.

Ei-muodolliset suojauskeinot sen sijaan soveltuvat hyvin myös keskeneräisten, vasta kehitysprosessin alkuvaiheessa olevien tuotteiden suojaamiseen, sillä suojauskeinojen käyttöön ei liity teostason tai keksinnöllisyyden kaltaisia kriteerejä<sup>375</sup>. Näin ollen ei-muodollisin keinoin voidaan suojata sellaisiakin kohteita, jotka eivät voisi saada esimerkiksi tekijän- tai patenttioikeudellista suojaa. Ensimmäisissä kokeiluissa työskennellään tyypillisesti vasta ideatasolla olevien ratkaisujen ja niitä havainnollistavien alkeellisten prototyyppien parissa, joten immateriaalioikeudellisen suojan saamisen edellytykset eivät välttämättä tällöin vielä täyty. Arvokasta – mutta ainakin vielä toistaiseksi patentti- ja tekijänoikeussuojan ulkopuolelle jäävää – ideaa voidaan tällöin pyrkiä suojaamaan ei-muodollisin keinoin.

---

372 Lahti 2006, s. 130–131

373 Hall ym. 2014, s. 389. Salaamisen kohdalla kuluja aiheuttavat salassapitosopimusten laatiminen, sisäisten salaamiskäytäntöjen luominen ja tiedon hallinnointi.

374 Päällysaho – Kuusisto 2006, s. 24

375 Samansuuntaisesti patentoinnin kannalta Hall ym. 2014, s. 388

Siinä missä patentointi näyttäytyy lähinnä vaihtoehtoisena edellä kuvatuille ei-muodollisille menetelmille<sup>376</sup>, soveltuu tekijänoikeus paremmin käytettäväksi yhdessä näiden suojauskeinojen kanssa. Edellä todetulla tavalla tekijänoikeussuoja on kuitenkin ohjelmiston kohdalla varsin rajallinen, joten tekijänoikeussuojan ulkopuolelle jäävien elementtien suojauksessa on turvauduttava muihin keinoihin. Tekijänoikeus suojaa ohjelmakoodia ja mahdollisesti myös tiettyjä muita ohjelmiston ulottuvuuksia (esim. graafinen käyttöliittymä voi saada suojaa kuvateoksena), mutta kattavan suojan saamiseksi erilaisia suojauskeinoja kannattaa käyttää täydentämään toisiaan. Kaikissa tapauksissa pätevää optimaalista suojauskeinojen yhdistelmää on mahdoton määrittää, joten sopiva yhdistelmä on aina riippuvainen yrityksen yksilöllisestä tilanteesta<sup>377</sup>.

## 7.3 Avoin innovaatio

### 7.3.1. Mitä avoimuudella tarkoitetaan?

Suljetulla innovaatiolla viitataan toimintatapaan, jossa painotetaan organisaation sisällä syntyviä ideoita ja organisaation itsenäisesti toteuttamaa ideoiden kaupallistamista. Perinteisesti yritykset ovat toteuttaneet tuotekehitystään suljetusti ja pyrkineet kontrolloimaan innovaatioitaan immateriaalioikeuksien avulla.<sup>378</sup> Sittenmin tiedon yhä nopeampi ja edullisempi saatavuus on kuitenkin mahdollistanut avoimen innovaation (*open innovation, OI*), jossa arvokaiden ideoiden nähdään voivan ilmaantua organisaation ulkopuoleltakin. Lisäksi ideoita ei välttämättä tarvitse kaupallistaa itse, vaan idea voi yhtä lailla kulkeutua markkinoille jotain ulkoista polkua pitkin (esimerkkinä lisensointi ulospäin). Kuitenkaan tarkoitus ei ole unohtaa omaa tuotekehitystä, vaan yhdistää organisaation omat kyvyt ja resurssit ulkopuolisten mahdollisuuksien paikantamiseen.<sup>379</sup> Yrityksen nähdään siis hyötyvän vuorovaikutuksesta suhteessa erilaisiin ulkopuolisiin toimijoihin, kuten asiakkaisiin, yliopistoihin, toisiin yrityksiin ja julkisiin instituutioihin<sup>380</sup>. On kuitenkin huomautettava, ettei avoimen innovaation oletettavasta kannattavuudesta ole päästy kirjallisuudessa täyteen yksimielisyyteen<sup>381</sup>.

376 Ks. esim. Holgersson – Wallin 2017. Salaaminen ei ole patentoinnin yhteydessä mahdollista, koska patentointiprosessissa keksintö tulee julkiseksi. Ennen patenttihakemuksen jättämistä tapahtuva keksinnön julkaiseminen puolestaan on patentoinnin edellytyksenä olevan uutuuden este. Edellä viitatuilla tavalla hidas patentointiprosessi vaikuttaa soveltuvan huonosti myös käytettäväksi yhdessä nopean innovoinnin kanssa.

377 Päällysaho – Kuusisto 2006, s. 2

378 Chesbrough 2006, s. 155

379 Chesbrough 2006, s. 43–44, Chesbrough – Appleyard 2007, s. 67 ja West – Gallagher 2006, s. 82

380 Aloini – Lazzarotti – Manzini – Pellegrini 2017, s. 1307

381 Aloini ym. 2017, s. 1310 ja Brem ym. 2017, s. 1288. Esimerkiksi tuoreessa tutkimuksessaan Brem ym. 2017 eivät löytäneet tukea hypoteesille, että avoimella innovaatiolla olisi ollut positiivinen vaikutus tutkimuksen pk-yritysten suoritukseen.



Ohjelmistoalalla tunnetuin esimerkki avoimesta innovaatiosta on avoin lähdekoodi. Kuitenkaan ohjelmistoalan avoin innovaatio ei tyhjenny pelkkään avoimen lähdekoodin käsitteeseen, eikä kaikki avoin lähdekoodi ole avointa innovaatiota. Westin ja Gallagherin mukaan avoin innovaatio edellyttää jonkinlaista liiketoimintamallia, eivätkä he katso GNU:n kaltaisia voittoa tavoittelemattomia avoimen lähdekoodin projekteja avoimeksi innovaatioksi. Vastaavasti esimerkiksi käyttäjien peleihin kehittämässä sisällössä on kyse avoimesta innovaatiosta, muttei avoimesta lähdekoodista.<sup>382</sup>

### **7.3.2 Avoin innovaatio ja suojakeinot**

Avoimen innovaation suhde immateriaalioikeuksiin on tietyllä tavalla ristiriitainen: toisaalta yksinoikeuksia ei haluta käyttää ulkopuolisten toimijoiden poissulkemiseen, mutta toisaalta turvallinen tiedon vaihtaminen ulkopuolisten kanssa edellyttäisi oman aineettoman omaisuuden suojaamista väärinkäytösten ehkäisemiseksi. Siinä missä eräät näkökulmat puoltavat joistain immateriaalioikeuksista luopumista, nähdään immateriaalioikeudet toisissa esityksissä avoimuuden mahdollistajina<sup>383</sup>. Avoimesta innovaatiosta voidaankin erottaa kaksi erilaista suhtautumista immateriaalioikeuksiin: 1) immateriaalioikeuksiin perustuva avoin innovaatio, jossa tietoa ei sinänsä salata ulkopuolisilta, mutta immateriaalioikeuksia perustetaan ja käytetään, ja 2) avoin innovaatio, jossa immateriaalioikeuksia ei perusteta tai niistä luovutaan<sup>384</sup>.

Aloinin ym. tutkimuksessa suosituin avoimeen innovaatioon liittynyt suojakeino oli salassapitosopimus. Myös liikesalaisuuksiin ja tiettyihin ei-muodollisiin suojauskeinoihin turvautttiin immateriaalioikeuksia useammin.<sup>385</sup> Patentti ei välttämättä ole avointa innovaatiota hyödyntävien yritysten suosituin suojamekanismi, mutta myös patentoinnin tärkeyden on katsottu kasvavan avoimen innovaation yhteydessä<sup>386</sup>. Mitään yksittäistä suojakeinoa on kuitenkin mahdoton yli muiden korostaa avoimen innovaationkaan suhteen, vaan myös avoimen innovaation kohdalla voidaan painottaa erilaisten suojakeinojen yhdistelemisen tärkeyttä: suojakeinoja yhdistelemällä saadaan käyttöön niiden erilaisia hyötyjä.<sup>387</sup>

<sup>382</sup> West – Gallagher 2006, s. 101–102

<sup>383</sup> Holgersson – Granstrand 2017, s. 1280, pitävät patentteja avoimen innovaation mahdollistajina pikemmin kuin avoimuuden hankaloittajina. Aloini ym. 2017, s. 1318–1319, katsovat oikeudellisten suojakeinojen ja niiden monimuotoisuuden vaikuttavan merkittävästi avoimuuteen ja mahdollistavan avoimen innovaation mallien omaksumisen. Henkel – Schöberl – Alexy 2014, s. 879–880, tuovat esiin sekä vahvojen immateriaalioikeuksien että joistain oikeuksista luopumisen etuja.

<sup>384</sup> Henkel ym. 2014, s. 880

<sup>385</sup> Aloini ym. 2017, s. 1313

<sup>386</sup> Patentoinnin suhteesta avoimuuteen ks. esim. Holgersson – Granstrand 2017, joiden tutkimuksessa liki kaikkien patentointimotiivien katsottiin olevan positiivisessa suhteessa avoimuuteen.

<sup>387</sup> Aloini ym. 2017, s. 1320–1321



Perustellulta vaikuttaa myös immateriaalioikeuksien ja avoimuuden suhteen yhdistäminen siihen, *miten* yritys käyttää immateriaalioikeuksia: yrityksen IPR-strategia voi joko hankaloida avoimuutta tai mahdollistaa sen. Kovin hyökkäävä omien oikeuksien valvominen voi pelottaa mahdolliset yhteistyötahot tiehensä, mutta immateriaalioikeudet voivat myös vetää puoleensa ulkopuolisia toimijoita, kun niitä käytetään strategisesti muuten kuin kontrollin välineenä (esim. neuvotteluissa tai innovatiivisen yrityskuvan viestittämisessä).<sup>388</sup>

### **7.3.3 Avoimuus kokeilussa**

Kokeilu voi olla osa suljettua innovaatioprosessia, vaikka idean toimivuutta jollain lailla testattaisiinkin käyttäjien reaktioita seuraamalla. Esimerkiksi kokeilu, jossa idean kiinnostavuutta arvioidaan tuotteen jonotuslistalle ilmoittautuneiden asiakkaiden määrän perusteella, ei merkitse ideoiden hankkimista tai niiden kaupallistamista yhdessä yrityksen ulkopuolisten tahojen kanssa. Melko tyypillistä lienee, että kokeilussa koehenkilöille esitetään yrityksen sisällä syntynyt idea, jota koehenkilöt voivat kommentoida, mutta he eivät osallistu tuotteen kehittämiseen ja muokkaamiseen esimerkiksi avoimen lähdekoodin käyttäjäyhteisöjen tavoin.

Jos käyttäjien kuitenkin toivotaan kokeilussa tapahtuvan vuorovaikutuksen kautta osallistuvan tuotekehitykseen aktiivisemmin ja heiltä tulevia ideoita saatetaan yrityksen toimesta kehittää jopa valmiiksi tuotteiksi asti, voi kyse olla avoimesta innovaatiosta. Käyttäjien tiiviimpi osallistaminen tuotekehitykseen edellyttää suurempaa avoimuutta tiedon kulun mahdollistamiseksi yrityksen ja käyttäjien välillä. Tällöin yrityksen on tehtävä valinta sen suhteen, haluaako se pitää kiinni immateriaalioikeuksistaan vai jossain määrin luopua niistä. Edellä kuvatulla tavalla immateriaalioikeudet voidaan nähdä joko avoimuuden mahdollistajina tai niistä luopumisen voidaan katsoa edistävän avoimuutta.

Suurempi avoimuus suhteessa kokeiluun osallistuviin käyttäjiin voi olla kannattava ratkaisu erityisesti silloin, kun kyse on käyttäjistä, jotka tiedostavat tuotteeseen liittyvät tarpeet muita nopeammin ja myös hyötyvät suuresti tarvettaan vastaavasta tuotteesta (ns. *lead users*). Tällaisilla käyttäjillä on usein arvokasta tietoa, mutta he eivät tyypillisesti vaadi ideoistaan immateriaalioikeuksia itselleen, vaan heitä motivoi jo toivo uusista, paremmista tuotteista.<sup>389</sup> Avoimuus suhteessa asiantunteviin käyttäjiin voi johtaa myös esimerkiksi OS-yhteisöjen yhteydessä todettuihin hyötyihin, kuten ulkoiseen avustukseen tuotekehi-

---

<sup>388</sup> Alexy – Criscuolo – Salter 2009, s. 71–73

<sup>389</sup> Felin – Zenger 2014, s. 922

tyksessä, testaamisessa, vikojen korjaamisessa ja ylläpidossa, joten pelkän kokeilussa saatavan palautteen ohella käyttäjiltä voidaan hakea muunkinlaista panosta. Toisaalta avoimuus on piirre, jota käyttäjät yhä enemmän vaativat yrityksiltä, ja joka voi tuottaa yritykselle positiivista mainetta.<sup>390</sup> Käyttäjien arvostama avoimuus voikin olla motivoiva seikka, joka kannustaa käyttäjiä osallistumaan tuotteen kehittämiseen. Yrityksen hanakka omien immateriaalioikeuksiansa varjeleminen saatetaan puolestaan kokea vieraannuttavaksi.

Jos yritys päättäisi toiminnassaan toteuttaa avointa innovaatiota pyrkimättä saamaan yksinoikeutta kokeilussa syntyvään aineettomaan omaisuuteen, ei luvussa 6 käsiteltyä tekijyyteen liittyvää problematiikkaa tarvitsisi kaikilta osin huomioida. Luvussa 6 esitetiin koehenkilöille mahdollisesti syntyvien immateriaalioikeuksien siirtämistä kokeilua tuotekehityksessään hyödyntävälle yritykselle yksinkertaisilla vakiosopimuksilla, mutta avoimen mallin omaksunut yritys voisi pyrkiä hallinnoimaan tällaisten useiden yrityksen ulkopuolisten tekijöiden panoksesta syntyviä teoksia esimerkiksi avoimen lähdekoodin lisenssein. Yrityksen omien ohjelmien lisensiointi avoimena lähdekoodina saattaisi osaltaan jouhevoittaa myös luvussa 5 käsiteltyä muiden tuottaman avoimen lähdekoodin hyödyntämistä osana omaa tuotetta<sup>391</sup>. Erilaiset avoimen lähdekoodin liiketoimintamallit<sup>392</sup> ovatkin hyvä esimerkki siitä, ettei ohjelmistotalan yrityksen toiminnan tarvitse perustua ohjelmistoon liittyvien immateriaalioikeuksien hallintaan.

Immateriaalioikeuksista luopumisena ymmärretyn avoimuuden voitaisiin ajatella siinä suhteessa keventävän tuotekehitysprosessia, että vaikealta ja kalliilta immateriaalioikeuksien valvomiselta ja puolustamiselta välttyään<sup>393</sup>. Toisaalta yksilöivän ja uniikin idean avoin jakaminen voi merkitä ehkä mittavienkin tulojen menetystä, joten selkeän kilpailuedun luoviin elementteihin yrityksen kannattaa pyrkiä saamaan yksinoikeus<sup>394</sup>. Yrityksen ei kuitenkaan tarvitse tehdä ehdotonta valintaa suljetuuden ja avoimuuden välillä, vaan yrityksen strategia voi puhtaasti avoimen tai suljetun innovaation sijasta olla ns. hybridimallia. Esimerkiksi ohjelmistoyrityksen koodista osa voi olla OS-yhteisöjen tuottamaa ja osa yrityksen työntekijöiden tuottamaa.<sup>395</sup> Käyttäjiä kokeilujen välityksellä aktivoivan yrityksenkään ei siis ole välttämätöntä luopua kaikista yksinoikeuksistaan, mutta käyttäjien kanssa tapahtuvasta vuorovaikutuksesta saatavan hyödyn maksimoimiseksi harkittu avoimuuden lisääminen voi olla kannattavaa.

390 Henkel ym. 2014, s. 882–883

391 Avoimen lähdekoodin lisenssirikkomusten ja lisenssien yhteensopimattomuuden välttely vaikuttaakin osaltaan yritysten päätökseen noudattaa itsekin suurempaa avoimuutta. Usein päätökseen kuitenkin liittyy myös lisenssien virheellistä tulkintaa. Ks. Henkel ym. 2014, s. 883.

392 Ks. esim. West – Gallagher 2006

393 Ks. Chesbrough – Appleyard 2007, s. 67

394 Linäker – Munir – Wnuk – Mols 2017, s. 19–20, 26

395 Chesbrough – Appleyard 2007, s. 70

## 8. LOPPUPÄÄTELMÄT

Kokeilu voidaan järjestää lukuisin eri tavoin – lakipykälien ja käytettävissä olevien resurs-sien ohella rajana on lähinnä kokeilun toteuttajien luovuus. Koska kokeileminen voi pitää sisällään hyvin erilaisia tapoja kontaktoida käyttäjiä tuotetta koskevan palautteen kerää-miseksi, on mahdoton muotoilla kovin yleispätevää esitystä kokeiluun liittyvistä tekijän-oikeudellisista toimentasuosituksista. Samaan hengenvetoon voitaneen kuitenkin todeta, että nämä erityyppiset käytettävissä olevat kokeiluasetelma- ja prototyypivaihtoehdot antavat kokeilun järjestäjälle mahdollisuuden suunnitella kokeilu tekijänoikeudelliselta kannalta joko enemmän tai vähemmän riskittömäksi. Tekijänoikeudellisen näkökulman huomioiminen on siis tärkeää jo ennen kokeilun toteuttamista.

Vaikka kokeilun tulisi olla raskasta suunnittelua karttava prosessi, jonka suunta hahmot-tuu hiljalleen uusien syklien myötä, vaatii ohjelmistotuotannon tekijänoikeudellisen ulot-tuvuuden hallitseminen kokeilussakin ennakkointia ja suunnittelua. Kokeilun suunnitte-luvaiheessa on tekijänoikeudelliselta kannalta keskeistä ratkaista, millaista prototyyppiä kokeilussa käytetään, millainen joukko koehenkilöitä tilanteeseen osallistetaan, millaisia tehtäviä koehenkilöille annetaan ja millä tavalla koehenkilöiden panosta hyödynnetään valmiissa tuotteessa. Koska kokeilusyklien kuluessa suositellaan vaihdettavan hyödynnet-tävää kokeiluasetelmaa ja prototyyppiä, kannattaa tekijänoikeudellisiakin seikkoja tarkas-tella uudelleen syklien välillä.

Ohjelmistokehitykseen pyrkivän kokeilun keskiössä oleva prototyyppi on teoskynnyksen ylittäessään tekijänoikeussuojan alainen teos, jonka tekijänoikeuden haltijalla on yksin-oikeus käyttää TekijäL 2 §:n määrittelemiä tekijän taloudellisia oikeuksia. Prototyypin luomiseen saatetaan myös käyttää jonkin kolmannen tahon tekijänoikeuden alaista ma-teriaalia. Jos prototyypin luomisessa on käytetty toisen tuottamaa teosta tai sen osaa, ku-ten lähdekoodia, on kyse TekijäL 2.2 §:n mukaisesta kappaleen valmistamisesta. Kokeilun toteuttamistavasta riippuen kokeilu voi merkitä myös TekijäL 2.3 §:n mukaista teoksen saattamista yleisön saataviin. Sen suhteen, onko kokeilu katsottavissa teoksen yleisön saa-taviin saattamiseksi, keskeistä on se, muodostaako kokeiluun osallistuvien henkilöiden joukko avoimen vai suljetun piirin.

Jonkin ulkopuolisen tahon tuottaman materiaalin hyödyntäminen osana prototyypin luo-mista voi olla houkutteleva vaihtoehto, sillä kaiken tekeminen alusta alkaen itse voi olla

liian raskas prosessi nopeaa ja resurssitehokasta kokeilua silmällä pitäen. Alkuvaiheen kokeilut voidaan viedä läpi yksinkertaisin koodittomin prototyypin, jotka tekijänoikeudellisessa mielessä eivät ole tietokoneohjelmia (mutta voivat saada suojaa jonain toisena teostyyppinä). Tällöin voidaan vielä välttyä lisensioimiseen liittyviltä kuluilta ja riskeiltä. Kuitenkin myöhemmissä kokeilusykleissä tavoiteltavan palautteen saamiseksi tarvitaan jo erilaista prototyyppiä, joista kehittyneimmät voivat sisältää suuriakin määriä koodia.

Nopeutta vaativissa ohjelmistokehitysprojekteissa käytetään nykyään yleisesti yrityksen ulkopuolelta hankittavia komponentteja, joita yhdistelemällä voidaan luoda laajojakin systeemejä verrattain vähällä vaivalla. Lisensioinnista aiheutuvia transaktiokustannuksia voidaan pyrkiä minimoimaan käyttämällä avoimen lähdekoodin komponentteja, joita on varsin matalalla kynnyksellä hankittavissa Internetistä. Parhaimmassa tapauksessa avoimen lähdekoodin hyödyntäminen voi mahdollistaa viimeistelyjen ominaisuuksien integroimisen prototyyppiin lähes ilmaiseksi, mutta oikeudellisesta näkökulmasta avoimen lähdekoodin käyttäminen edellyttää huolellisuutta. Olennaista on ymmärtää ero sallivien ja vastavuoroisuutta vaativien lisenssien välillä, sillä lisenssien sisältämät vastavuoroisuusvelvoitteet voivat hankaloittaa erilaisten lisenssien alaisten komponenttien yhdistelyä ja kokonaisuuden uudelleen lisensioimista. Lisäksi avoimen lähdekoodin lisensseissä on lähes poikkeuksetta vastuunrajoituslauseke, jolla lisenssinsaajalle siirretään vastuu lisensioitavaan ohjelmaan sisältyvien teknisten virheiden ja kolmannen immateriaalioikeuksien loukkausten aiheuttamista vahingoista. Riskejä voidaan pyrkiä hallitsemaan lisensioitavien komponenttien koodin alkuperän selvittelyllä, yrityksen sisäisin avointa lähdekoodia koskevin ohjeistuksin ja systeemin arkkitehtuurin tarkalla suunnittelulla. Vaikka avoimen lähdekoodin taustalla on ajatus innovaation edistämisestä yhteisresursseja ylläpitämällä, voi erityisesti vastavuoroisuutta vaativien lisenssien alaisten teosten hyödyntäminen olla kaupallisista toimijoista epäilyttävää. Osin vastavuoroisuusvelvoitteisiin liittyvä ”virusvirkutuksen” pelko tosin johtuu vaikeaselkoisten lisenssien tulkintaongelmista.

Koska kokeilussa on kyse ryhmätyöstä, joka kokeilujen syklisyyden vuoksi on tyypillisesti sekä samanaikaista että peräkkäistä (horisontaalinen ja vertikaalinen yhteistyö), voi tekijänoikeus testattavaan ohjelmistoon syntyä useille työskentelyyn osallistuville henkilöille yhteisesti. Kokeiluun osallistuu kehitystiimin lisäksi yrityksen ulkopuolisia koehenkilöitä, joiden luova panos voi ainakin teoriassa mutkistaa kokeiluun liittyviä tekijänoikeudellisia suhteita. Jos koehenkilö vain osallistuu testaamiseen ja antaa prototyypistä palautetta, ei

hänen toimintansa yleensä ole tekijänoikeudellisesti merkityksellistä. Jos koehenkilö sen sijaan voi ottaa kokeilussa aktiivisemman roolin ja osallistua myös prototyypin muokkamiseen, saattaa hänelle syntyä tekijänoikeus panokseensa. Koska tietokoneohjelman tekijänoikeussuoja rajoittuu lähinnä ohjelmakoodiin, edellyttäisi tietokoneohjelman tekijäksi tuleminen koehenkilöltä käytännössä osallistumista koodin luomiseen. Lienee epätodennäköistä, että tavallisella käyttäjällä olisi tällaista erityisosaamista tai että hänelle annettaisiin tilaisuus päästä käsiksi lähdekoodiin. On kuitenkin huomattava, että ohjelmistoon voi kuulua muitakin teoslajeja, joiden luomiseen koehenkilöllä saattaisi olla mahdollisuus osallistua.

Jotta yritys voisi vapaasti hyödyntää aineetonta omaisuuttaan, kannattaa immateriaalioikeuksien olla yrityksen nimissä. Tekijänoikeus syntyy aina luonnolliselle henkilölle, mutta työsuhteessa syntyneet tekijänoikeudet voivat siirtyä työnantajayritykselle TekijäL 40 b §:n tai normaalikäytösäännön nojalla. Jos työsuhteen tunnusmerkit täyttyvät, voi koehenkilön ja yrityksen välillä olla hiljainen työsopimus, vaikka asiasta ei nimenomaisesti olisi sovittu. Kaikki tunnusmerkit – etenkin vastikkeellisuus- ja direktiotunnusmerkit – tuskin tavallisesti kuitenkaan täyttyvät. Työsuhteen ulkopuolella tekijänoikeudet siirtyvät vain sopimalla, joten tekijänoikeuksien siirtäminen voi olla syytä järjestää koehenkilöiden kanssa yksinkertaisin vakiosopimuksin. Toinen vaihtoehto on määritellä koehenkilöiden rooli kokeilussa sellaiseksi, että he eivät osallistu tekijänoikeuden alaan kuuluvan tuotteen ilmaisumuodon luomiseen, vaan tekijöitä ovat vain kehitystiimiin kuuluvat yrityksen työntekijät.

Tekijänoikeus on vain osa yrityksen IPR-strategiaa, sillä kattavin suoja saadaan erilaisia suojauskeinoja yhdistelemällä. Tekijänoikeus ja ei-muodolliset suojauskeinot voivat olla helpommin ja joustavammin käytettävissä nopeiden kokeilujen yhteydessä kuin hitaan ja resursseja vaativan rekisteröimisprosessin kautta saavutettava patentti, mutta patenttiin liittyy toisaalta tiettyjä strategisia etuja, joita tekijänoikeus ja ei-muodolliset suojauskeinot eivät tarjoa. Lisäksi patentilla voidaan suojata ohjelmistoon liittyviä ratkaisuja, jotka eivät kuulu tekijänoikeuden piiriin. Halukkuus patentointiin on kuitenkin syytä tiedostaa jo ennen keksinnön testaamista kokeilussa, sillä väärin toteutettu kokeilu voi merkitä patentoitavalta keksinnöltä edellytettävän uutuuden menetystä. Ei-muodollinen suojaaminen puolestaan on keskeistä erityisesti kehitysprosessin varhaisimmissa kokeiluissa, joissa työskennellään vielä tekijän- ja patenttioikeudellisen suojan ulkopuolelle jäävien ideoiden parissa.

Perinteisesti IPR-strategia on perustunut ajatukseen aineettoman omaisuuden yksinoikeudella tapahtuvasta hyödyntämisestä. Liiketoiminnan ei kuitenkaan tarvitse perustua immateriaalioikeuksien omistamiseen, vaan innovaation tehostamiseksi oikeuksista voidaan osittain myös luopua. Avoimen innovaation piirissä immateriaalioikeudet on nähty sekä innovaatiota kangistavina että avoimuuden mahdollistavina tekijöinä. Jos kokeilussa halutaan hyödyntää käyttäjien ideoita ja osaamista laajemmin kuin vain palautteen tiedustelun muodossa, voi suurempi avoimuus mahdollistaa paremman informaation kulkeamisen yrityksen ja käyttäjien välillä. Yrityksen innovaatiotoiminnan vilkkauden kannalta kriittiset seikat vaikuttaisivat liittyvän pikemminkin yrityksen tapaan käyttää immateriaalioikeuksia kuin immateriaalioikeusjärjestelmän olemassaoloon sinänsä. Myös tekijänoikeutta voidaan toisaalta käyttää muut poissulkevana kontrollin välineenä, mutta toisaalta se voi ehkäistä käyttäjien kanssa tapahtuvaan yhteisluomiseen liittyviä riskejä ja siten mahdollistaa tiiviimmän yhteistyön ulkopuolisten tahojen – kuten kokeiluun osallistuvien käyttäjien – kanssa.

Jatkotutkimusta ajatellen kiinnostavia tekijänoikeudellisia kysymyksiä liittyy selkeästi myös muihin metodeihin, joilla käyttäjiä osallistetaan tuotekehitykseen. Kokeilun toteuttamistavasta riippuen koehenkilöllä ei välttämättä ole minkäänlaista luovaa roolia kokeilussa, mutta toisilla menetelmillä nimenomaan pyritään käyttäjien luovien ideoiden löytämiseen ja kaupalliseen hyödyntämiseen, jolloin käyttäjillekin voi todennäköisemmin syntyä kehitettävään tuotteeseen kohdistuvia immateriaalioikeuksia. Tässä tutkielmassa on lyhyesti käsitelty myös patenttia, mutta patenttia koskeva esitys on vain suppea johdatus kokeilujen patenttioikeudellisiin näkökohtiin. Koska patentoinnilla on ohjelmistoalalla yhä kasvava merkitys, voisi tarpeellista olla tarkastella ohjelmistopatenttiakin suhteessa kokeilemiseen tai muihin ketteriin menetelmiin.